# Oracle Fusion Cloud Student Financial Planning

## How do I use the Generative AI Financial Aid Automation Configuration Toolkit?

Oracle Fusion Cloud Student Financial Planning
How do I use the Generative AI Financial Aid Automation Configuration Toolkit?

G29535-01

*Copyright* © 2025, Oracle and/or its affiliates.

Author: Higher Education Information Development Team

# Contents

**Get Help** i

**1 Overview of the Generative AI Financial Aid Automation Configuration Toolkit** 1

Overview of the Generative AI Financial Aid Automation Configuration Toolkit 1

**2 Before You Begin** 3

**3 Student Financial Planning Groovy Scripts and Prompts** 5

Prompt Structure 8
ISIR Processing Scripts 15
Financial Aid Calculation Scripts 251
Enrollment and Academic Scripts 357
Document Management Scripts 409
Nonfederal (NFR) Resources Scripts 445
Satisfactory Academic Progress (SAP) Scripts 576
Disbursement Scripts 594
Miscellaneous Scripts 661

**4 Troubleshoot Common Issues** 747

Troubleshoot Common Issues 747

# Get Help

There are a number of ways to learn more about your product and interact with Oracle and other users.

## Get Help in the Applications

Some application pages have help icons to give you access to contextual help. If you don't see any help icons on your page, click your user image or name in the global header and select Show Help Icons. If the page has contextual help, help icons will appear.

## Get Support

You can get support at *My Oracle Support*. For accessible support, visit *Oracle Accessibility Learning and Support*.

## Get Training

Increase your knowledge of Oracle Cloud by taking courses at *Oracle University*.

## Join Our Community

Use *Cloud Customer Connect* to get information from industry experts at Oracle and in the partner community. You can join forums to connect with other customers, post questions, suggest *ideas* for product enhancements, and watch events.

## Learn About Accessibility

For information about Oracle's commitment to accessibility, visit the *Oracle Accessibility Program*. Videos included in this guide are provided as a media alternative for text-based topics also available in this guide.

## Share Your Feedback

We welcome your feedback about Oracle Applications user assistance. If you need clarification, find an error, or just want to tell us what you found helpful, we'd like to hear from you.

You can email your feedback to *oracle_fusion_applications_help_ww_grp@oracle.com*.

Thanks for helping us improve our user assistance!

# 1 Overview of the Generative AI Financial Aid Automation Configuration Toolkit

This toolkit provides prompts that, when used with a generative AI tool, help automatically add clear, structured comments to your Groovy scripts.

Maintaining Groovy scripts can be difficult, especially when they lack clear explanations. Without proper documentation, making updates can feel risky, and understanding how these scripts interact with Oracle Fusion Cloud Student Financial Planning (SFP) can be overwhelming. Many teams struggle with keeping their scripts organized and easy to understand, which can lead to mistakes and slow down progress.

This documentation provides information on how to use these simple yet powerful prompts that automatically add clear, structured comments to your Groovy scripts. These comments explain how different parts of the script work, include relevant SFP rules, and ensure consistency across all scripts. With better documentation, teams can quickly understand and update scripts with confidence, reducing errors and making collaboration easier. By using these prompts with a generative AI tool, you can transform complex scripts into well-documented, easy-to-maintain tools that support your team's success.

The generative AI financial aid automation configuration toolkit gives you powerful prompts that:

- Integrate official REST API documentation. This means important SFP REST API information is automatically added, including clear explanations of how different parts of the script work and when to use them. Platform-specific business rules and validations are also included.
- Enforce standardized documentation. There's a consistent comment structure across all script types. There's also a unified format for conditions, boundaries, and calculations. Because of these, scripts become easier to understand regardless of complexity.
- Enhance maintainability. Because there's a consistent comment structure and uniform formats for identifying information, you can quickly understand a script's function. And when you understand what a script does, it makes it easier and safer to update the script if you need to. You also minimize errors because validation requirements are available and are explained. With this information, knowledge transfer between team members becomes easier.

# 2 Before You Begin

Here are a few things you need to do to ensure the best results when using the generative AI financial aid automation configuration toolkit.

## Choose a Generative AI Tool

There's a wide variety of generative AI tools out there—from powerful public models to flexible open-source options. Choosing the right one can make a big difference, so it's worth taking the time to explore what fits best with your goals. Think about what matters most to you—whether it's accuracy, how easily it integrates with your applications, or how much control you want over the results.

We recommend using OpenAI's ChatGPT 4o and 4o-mini models. These models have consistently shown exceptional performance for generating commented Groovy code and handling complex prompts. The ChatGPT 4o and 4o-mini models provide clear, maintainable, and context-aware outputs, making them ideal for this task.

If you're already using a generative AI provider or exploring open-source solutions, feel free to choose the one you're most comfortable with and that delivers the best results for your use case. Key factors to consider when selecting an LLM include:

- Model performance. You need to evaluate the quality of generated outputs, especially for structured tasks like code commenting and reviews.
- Cost. Consider the pricing model and its alignment with your budget.
- Data privacy. Verify the provider's compliance with your organization's data security and privacy requirements.

By choosing the appropriate tool, you can maximize the efficiency and effectiveness of your Groovy script development process.

## Prepare Your Code

Ensure that your code is properly formatted. Make sure you remove sensitive data. Lastly, verify that the script type matches the prompt.

## How to Use the Prompts

When using the prompts, make sure you don't change the context sections: it's important to keep original prompt structure intact.

Replace only this code section (as indicated in the prompt): `## Groovy Script for Analysis`

# How to Review the Generated Documentation

When reviewing the generated documentation, check that all sections are complete. Make sure you also check for correct method references. Also make sure that the business logic is correctly explained.

# 3 Student Financial Planning Groovy Scripts and Prompts

Here's a list of all available Oracle Student Financial Planning Groovy scripts, organized by functional category. Each script includes its name, purpose, return type, and a ready-to-use prompt that you can copy into your generative AI tool to generate well-documented code.

The scripts are grouped into categories to help you quickly find what you need.

### *ISIR Processing Scripts*

| Script Name | Purpose | Return Type |
|---|---|---|
| isirUsedInPackaging | Select the ISIR that should be used in packaging for an Award Year. | IIsirRecordAPI |
| isirCCodeAutoCodeClearingLogic | Determine if an ISIR c-code is auto cleared. | Boolean |
| isirAssumptionAutoCodeClearingLogic | Determine if an ISIRassumption is auto cleared. | Boolean |
| isirHighlightAutoCodeClearingLogic | Determine if an ISIRhighlight is auto cleared. | Boolean |
| isirRejectCodeAutoCodeClearingLogic | Determine if an ISIR reject code is auto cleared. | Boolean |
| isirVerificationClearanceRules | Determine if an ISIRverification is auto cleared. | Boolean |
| isirUseSubsequentIsir | Define which fields on ISIR will trigger an ISIR review when subsequent ISIRs enter SFP for a Student. | Boolean |
| isirDiscrepancyEvaluation | Determinewhether an ISIR correction for a given ISIR field is required. | Void |
| isirHighlightsDocumentsRequired | Determine the documents required to resolve an ISIR highlight. | Void |
| isirCCodeDocumentsRequired | Determine the documents required to resolve an ISIR c-code. | Void |
| isirRejectCodeDocumentsRequired | Determine the documents required to resolve an ISIR reject code. | Void |
| isirVerificationDocumentsRequired | Determine the documents required to resolve an ISIR verification. | Void |
| isirAssumptionDocumentsRequired | Determine the documents required to resolve an ISIR assumption. | Void |
| isirDiscrepancyRequiredDocuments | Determine whether the isirDiscrepancyEvaluation script should be executed. | Void |
| isirDiscrepancyAdditionalDocumentsRequired | Determine whether additional documentation should be collected. | Void |

### *Financial Aid Calculation Scripts*

| Script Name | Purpose | Return Type |
|---|---|---|
| gradeLevelProgression | Calculate the grade level. | Integer |
| awardOverridingCriteria | Override award amounts. | IAwardOverrideAPI |
| awardYearSelectionCriteria | Determine the award year to use in crossover payment periods. | IScriptedCrossoverArbitratorResult |
| plusCreditDecisionMatchingCriteria | Determine the (G)PLUSloans applicable to the term or payment period. | List<IPlusLoanAPI> |
| medicalAmount | Determine the medical amount. | BigDecimal |
| pellEnrollmentIntensity | Determine the Enrollment Intensity percentage for the Pell award. | BigDecimal |

### *Enrollment and Academic Scripts*

| Script Name | Purpose | Return Type |
|---|---|---|
| enrollmentStatusOverride | Determine whether the calculated enrollment status should be overridden. | String |
| globalEnrollmentStatusCriteria | Specifies how the application should find the student's enrollment status. | String |
| summerTermType | Defines if the Summer Term should be treated as Header or Trailer. | String |
| nonTermAcyMonths | Determine the number of months in a nonterm academic year. | Integer |
| termPaymentPeriodMonths | Determine the number of months in a term payment period. | BigDecimal |

### *Document Management Scripts*

| Script Name | Purpose | Return Type |
|---|---|---|
| documentOwnerLogic | Determine the document owners. | Void |
| documentFaAcceptance | Determine the Financial Aid acceptance of a document. | Boolean |
| documentCancellation | Determines when a document request can be canceled. | Void |
| documentMetadataAdditionalResolutionAction | Evaluate document metadata fields for document acceptance. | Void |

### *Satisfactory Academic Progress (SAP) Scripts*

| Script Name | Purpose | Return Type |
|---|---|---|
| sapDisablePaceOfCompletion | Determines if Pace of Completion calculations are needed for SAP. | Boolean |
| sapEvaluationPeriodDetermination | Determines if SAP Period needs evaluation at the end of the academic year. | Boolean |

### *Disbursement Scripts*

| Script Name | Purpose | Return Type |
|---|---|---|
| disbursementDates | Evaluates anticipated disbursement dates. | List<LocalDate> |
| configurableDisbursementCriteria | Evaluates configurable disbursement criteria. | Boolean |
| configurableLateDisbursementCriteria | Evaluates configurable late/pwd disbursement criteria. | Boolean |
| originationCriteria | Evaluates origination criteria for a disbursement. | Boolean |

### *Miscellaneous Scripts*

| Script Name | Purpose | Return Type |
|---|---|---|
| letterCodeCriteria | Determines the letter code for given parameters. | String |
| notificationRequiredCriteria | Determines if a student notification must be triggered. | Boolean |
| coaCostValue | Calculates the value of a COA item. | BigDecimal |
| additionalCoaComponents | Evaluates if a COA configuration is applicable. | Boolean |
| exitCounselingGraduatingCriteria | Evaluates exit counseling criteria for graduating students. | Boolean |
| exitCounselingWithdrawnCriteria | Evaluates exit counseling criteria for withdrawn students. | Boolean |
| professionalJudgementInitiatingConditions | Determines if a student meets the criteria for professional judgement. | Boolean |
| professionalJudgementRequiredDocuments | Determines the documents required for professional judgement. | Void |

# Prompt Structure

The prompts are structured in a specific way to provide the generative AI tool with the script context, available methods and bindings, and identifies the documentation requirements.

## Script Context

This section indicates:

- Script purpose and functionality
- Integration points with the Student Financial Planning
- Compliance requirements
- The business impact

## Available Methods and Bindings

In this section, these elements are indicated:

- Platform-specific methods
- Integration points
- Required validations
- Business rules

## Documentation Requirements

And in this section, these elements are indicated:

- File-level documentation format
- Section-level documentation structure
- Comment standards and expectations
- Required aspects (conditionals, boundary checks, arithmetic)

## Example Output

Here's an example of the output that the generative AI tool produces if you were to copy and paste the prompt for GradeLevelProgression (financial aid calculation script xref).

```
//file:noinspection UnnecessaryQualifiedReference

@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.GradeLevelProgressionScript)

package CHANGE_ME // this should be updated to your actual package name


import com.oracle.sfp.scripting.api.*

import com.oracle.sfp.scripting.api.util.*


/**
```

```
* Script Name: gradeLevelProgression

* Version: 1.0

*

* Primary Purpose and Functionality:

* - Calculates the student's grade level based on completed credits, program type, and academic year
progression.

* - Ensures compliance with institutional and federal financial aid requirements.

*

* Key Calculations:

* - Summation of completed credits from prior academic years and current academic year.

* - Division of total credit units by academic year duration units (AYD).

* - Grade level determination based on credit thresholds and program type.

*

* Critical Business Rules Enforced:

* - Credits earned prior to the program start date are appropriately included or excluded.

* - Certificate and Associate programs have unique grade level limits.

* - Financial aid regulations and academic progress requirements are enforced.

*

* Boundary Conditions:

* - Actual start date and academic year start date must be valid.

* - Completed credit calculations must exclude canceled periods.

*

* Financial Aid Implications:

* - Grade level directly impacts loan eligibility and Pell grant disbursement limits.

*

* Required Inputs:

* - Program type, total completed units, academic year units, and start dates.

*

* Expected Outputs:

* - An integer representing the grade level.

*

* Error Handling:

* - Logs debug information for each calculation step.
```

```
 * - Ensures invalid or null data is handled gracefully.

 *

 * Integration Points:

 * - Relies on methods from acy, acys, program, and term bindings.

 *

 * Compliance Considerations:

 * - Ensures compliance with federal aid requirements and academic progress policies.

 */


// CONDITIONALS:

// Check 1: Reference Date Determination

// Variables: program.isTerm(), term.getStartDate(), acy.getStartDate()

// Purpose: Determine the reference date based on whether the program is term-based.

// Logic:

// - If the program is term-based, use the term start date.

// - Otherwise, use the academic year start date.

// Success Path: Assigns the correct reference date for calculations.

// Failure Path: N/A

// Business Rules: Reference date is critical for filtering credits in current academic year.

// Error Handling: N/A

// Financial Impact: Ensures credits are correctly included in financial aid eligibility.

def referenceDate = program.isTerm() ? term.getStartDate() : acy.getStartDate();


// ARITHMETIC:

// Operation 1: Total Completed Units from Previous Academic Years

// Formula: sum(acys.findAll { it.getNumber() < acyNo }.collect { it.getStatus() == "ACTUAL" ?
 it.getCompletedUnits() : it.getUnits() })

// Variables:

// - acys: List of academic years.

// - acyNo: Current academic year number.

// Steps:

// 1. Filter academic years with a number less than the current year.

// 2. Sum completed units for "ACTUAL" status or total units otherwise.

// Business Context: Ensures all prior academic year credits are counted.

// Result: Total units from previous academic years.
```

```
// Financial Impact: Impacts progression and loan eligibility.

def totalPrevAcyUnits = acys

 .findAll { it.getNumber() < acyNo }

 .collect { it.getStatus() == "ACTUAL" ? it.getCompletedUnits() : it.getUnits() }

 .sum(0.0);


// CONDITIONALS:

// Check 2: Valid Loan Payment Periods

// Variables: acy.getLoanPaymentPeriods(), referenceDate

// Purpose: Include only valid loan payment periods in the current academic year.

// Logic:

// - Exclude canceled payment periods.

// - Include periods with a start date before the reference date.

// Success Path: Filters and includes valid loan payment periods.

// Failure Path: Excludes invalid periods.

// Business Rules: Only valid payment periods affect financial aid calculations.

// Error Handling: Skips invalid periods and logs any issues.

// Financial Impact: Impacts total credits counted for grade level calculation.

def totalCurAcyUnits = acy.getLoanPaymentPeriods()

 .findAll { it.getStatus() != "CANCELED" && it.getStartDate() < referenceDate }

 .collect { it.getUnits() }

 .sum(0.0);


// ARITHMETIC:

// Operation 2: Total Credits Completed Before Program Start Date

// Formula: sum(program.getCourses().findAll { conditions }.collect { it.getUnits() })

// Variables:

// - program.getCourses(): List of courses.

// - actualStartDate: The program's actual start date.

// Steps:

// 1. Filter courses that are passed and completed before the program start date.

// 2. Sum their credit units.

// Business Context: Ensures pre-program credits are correctly included.

// Result: Total credits earned before program start date.
```

```
// Financial Impact: Ensures accurate grade level determination.

def totalCreditsCompletedBeforeProgramStartDate = program.getCourses()

 .findAll {

 it.getSchedulingStatus() == "PASSED" &&

 it.getEndDate() < actualStartDate &&

 (it.getIncompleteResolutionDate() == null || it.getIncompleteResolutionDate() < actualStartDate)

 }

 .collect { it.getUnits() }

 .sum(0.0);


// ARITHMETIC:

// Operation 3: Total Credit Units

// Formula: sum(transfer units, assessed units, previous academic year units, current academic year units,
 pre-program units)

// Variables:

// - program.getTotalAcceptedTransferUnits()

// - program.getAssessedUnits()

// - totalPrevAcyUnits

// - totalCurAcyUnits

// - totalCreditsCompletedBeforeProgramStartDate

// Steps:

// 1. Add all sources of credit units.

// 2. Store the result in totalCreditUnits.

// Business Context: Provides the total units for grade level calculation.

// Result: Total completed credit units.

// Financial Impact: Determines eligibility for grade level progression.

def totalCreditUnits = (program.getTotalAcceptedTransferUnits() +

 program.getAssessedUnits() +

 totalPrevAcyUnits +

 totalCurAcyUnits +

 totalCreditsCompletedBeforeProgramStartDate);


// ARITHMETIC:

// Operation 4: Credits Per Academic Year Duration (AYD) Unit
```

```
// Formula: totalCreditUnits / program.getAydUnits()

// Variables:

// - totalCreditUnits

// - program.getAydUnits(): Units representing academic year duration.

// Steps:

// 1. Divide total credits by AYD units.

// Business Context: Normalizes credits to academic year duration for progression.

// Result: Credits per AYD unit.

// Financial Impact: Impacts grade level calculation for financial aid.

def creditPerAydUnit = totalCreditUnits / program.getAydUnits();


// CONDITIONALS:

// Check 3: Base Grade Level Determination

// Variables: program.isGraduate(), program.getProgramType()

// Purpose: Determine the base grade level based on program type.

// Logic:

// - Graduate programs start at grade level 6.

// - Post-baccalaureate certificate programs start at grade level 5.

// - Other programs start at grade level 1.

// Success Path: Assigns the correct base grade level.

// Failure Path: N/A

// Business Rules: Program type dictates initial grade level.

// Error Handling: N/A

// Financial Impact: Ensures accurate starting point for progression.

def baseGradeLevel = (program.isGraduate()

 ? 6

 : program.getProgramType() == "CERTIFICATE_POST_BACCALAUREATE"

 ? 5

 : 1);


// ARITHMETIC:

// Operation 5: Unbounded Grade Level

// Formula: baseGradeLevel + creditPerAydUnit.intValue()

// Variables:
```

```
// - baseGradeLevel

// - creditPerAydUnit

// Steps:

// 1. Add the integer value of credits per AYD unit to the base grade level.

// Business Context: Calculates progression without upper limits.

// Result: Unbounded grade level.

// Financial Impact: Determines progression threshold.

def unboundedGradeLevel = baseGradeLevel + creditPerAydUnit.intValue();


// CONDITIONALS:

// Check 4: Program Type and Upper Bound Grade Level

// Variables: program.getProgramType(), program.isUndergraduate()

// Purpose: Determine the maximum grade level based on program type.

// Logic:

// - Associate programs have a maximum grade level of 2.

// - Undergraduate programs have a maximum grade level of 5.

// - Graduate programs have a maximum grade level of 7.

// Success Path: Assigns the correct upper bound grade level.

// Failure Path: N/A

// Business Rules: Grade level limits depend on program type.

// Error Handling: N/A

// Financial Impact: Ensures compliance with financial aid policies.

def isAssociate = program.getProgramType() == "ASSOCIATE";

def gradeLevelUb = (isAssociate

 ? 2

 : program.isUndergraduate()

 ? 5

 : 7);


// CONDITIONALS:

// Check 5: Final Grade Level Calculation

// Variables: unboundedGradeLevel, gradeLevelUb

// Purpose: Determine the final grade level based on upper bounds.

// Logic:
```

```
// - Take the minimum of the unbounded grade level and the upper bound.

// Success Path: Assigns the final grade level.

// Failure Path: N/A

// Business Rules: Grade level must not exceed program-specific limits.

// Error Handling: N/A

// Financial Impact: Ensures accurate financial aid reporting.

def gradeLevel = [unboundedGradeLevel, gradeLevelUb].min();


// Logging all calculated variables for traceability

log.debug("actualStartDate={}, referenceDate={}, totalPrevAcyUnits={}, totalCurAcyUnits={},
 totalCreditsCompletedBeforeProgramStart
```

# ISIR Processing Scripts

These are the prompts for scripts related to Institutional Student Information Record (ISIR) processing, including verification, corrections, and document requirements.

## ISIR Used in Packaging

Name: isirUsedInPackaging

Purpose: Selects the ISIR that should be used in packaging for an Award Year.

In Baseline Config: Yes

Return Type: IIsirRecordAPI

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "isirUsedInPackaging"
Primary Function: Selects the ISIR that should be used in packaging for an Award Year
Expected Output Type: IIsirRecordAPI
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
```

```
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
awy
Purpose: The award year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
awyPeriod
Purpose: The period of the award year of the funding element being evaluated.
Type: IAwardYearAPI
Available Methods:
* asList
 - Returns: IAwardYearsAPI
 - Function: Returns an IAwardYearsAPI instance containing only this Award Year.'
* getEndYear
 - Returns: int
 - Function: Gets the end year of this Award Year, for example 2023 for Award Year 2022-2023.'
* getNext
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that follows this one chronologically.'
* getPrevious
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that precedes this one chronologically.'
* getStartYear
 - Returns: int
 - Function: Gets the start year of this Award Year, for example 2022 for Award Year 2022-2023.'
* getYear
 - Returns: int
 - Function: Short for getEndYear().'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
```

```
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
No methods available
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
```

```
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
```

```
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
```

```
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
```

```
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
```

```
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
```

```
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
```

```
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
```

```
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
```

```
    - Returns: Boolean
    - Function: Gets the financial aid request status.'
  * getFirstName
    - Returns: String
    - Function: Gets the first name.'
  * getGender
    - Returns: String
    - Function: Gets the gender.'
  * getHighSchoolEquivalency
    - Returns: Boolean
    - Function: Gets the high school equivalency status.'
  * getLastName
    - Returns: String
    - Function: Gets the last name.'
  * getMiddleName
    - Returns: String
    - Function: Gets the middle name.'
  * getPersonId
    - Returns: Long
    - Function: Gets the person ID.'
  * getProgramType
    - Returns: String
    - Function: Gets the program type.'
  * getRecentFinancialAidHistoryRecords
    - Returns: IRecentFahRecordsAPI
    - Function: Gets the recent financial aid records.'
  * getSchoolCode
    - Returns: String
    - Function: Gets the school code.'
  * getSchoolEmployee
    - Returns: Boolean
    - Function: Gets the school employee status.'
  * getSchoolEmployeeDependent
    - Returns: Boolean
    - Function: Gets the school employee dependent status.'
  * getSsn
    - Returns: ISocialSecurityNumber
    - Function: Gets the SSN.'
  * getStudentId
    - Returns: Long
    - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
Available Methods:
  * getPeriodsIntersectingWithGroupForThePeriod
    - Returns: List<? extends IPeriod>
    - Function: Find the periods student was in the given group in the specified period of time'
  * isInGroupAtAnyPointInThePeriod
    - Returns: boolean
    - Function: Checks if student was in the given group in the specified period at any point of time.'
  * isInGroupAtDate
    - Returns: boolean
    - Function: Checks if student was in the given group on the given date.'
  * isInGroupDuringCompletePeriod
    - Returns: boolean
    - Function: Checks if student was in the given group in the specified period for the whole time.'
  * numberOfDaysInGroupForThePeriod
    - Returns: int
    - Function: Count number of days student was in the given group in the specified period of time'
  * getActiveGroupsAtAnyPointInThePeriod
    - Returns: List<String>
    - Function: Find the student's groups active at any point during specified period of time'
  * getActiveGroupsAtDate
    - Returns: List<String>
    - Function: Find the student's active groups at the given date'
```

```
* getActiveGroupsDuringCompletePeriod
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
 // Operation [number]: [name of calculation]
 // Formula: [complete mathematical expression]
 // Variables:
 // - [each variable and its purpose]
 // - [data types and units]
 // Steps:
 // 1. [detailed step-by-step breakdown]
```

```
 // 2. [intermediate calculations]
 // 3. [transformations applied]
 // Business Context: [why this calculation is needed]
 // Result: [meaning of the output]
 // Financial Impact: [effect on aid calculations]
 // Limitations: [any mathematical constraints]
 // If N/A, explicitly state: "ARITHMETIC: N/A"
ADDITIONAL REQUIREMENTS:
1. Complex Logic Sections:
 - Must break down multi-step operations
 - Explain interactions between components
 - Detail data flow and transformations
2. Error Handling:
 - Document all error conditions
 - Explain recovery procedures
 - Detail logging and reporting
3. API Interactions:
 - Document all platform method calls
 - Explain data transformations
 - Detail error handling
4. Financial Aid Impact:
 - Explain effects on aid calculations
 - Document regulatory compliance
 - Detail student eligibility impact
Example Section With All Aspects:
// CONDITIONALS:
// Check 1: Payment Period Status Validation
// Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
```

```
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.IsirUsedInPackagingScript)
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
//get all compatible ISIRs for the Award Year where no reject codes exist and a SAI exists
def awardYearISIRs =
 allAwardYearIsirs.forAwardYear(awy).getCompatibleWithPackaging().findAll{it.getRejectCodes().isEmpty() &&
 it.getField(IsirField.TXID_SAI).getOrNull() != null}
//if a Valid or Unverified ISIR exists, then return it
def validOrUnverifiedISIR = awardYearISIRs.find{["Valid", "Unverified"].contains(it.isirStatus)}
if (validOrUnverifiedISIR != null) {
 return validOrUnverifiedISIR
}
//if a Corrected ISIR exists, then return it if the ISIR had been previously Valid or Unverified
def previouslyValidOrUnverifiedCorrectedISIR = awardYearISIRs.find{it.isirStatus == "Corrected" && ["Valid",
 "Unverified"].contains(it.getIsirStatusBeforeCorrection())}
if (previouslyValidOrUnverifiedCorrectedISIR != null) {
 return previouslyValidOrUnverifiedCorrectedISIR
}
//if a Corrected ISIR exists, then return it
def correctedISIR = awardYearISIRs.find{it.isirStatus == "Corrected"}
if (correctedISIR != null) {
 return correctedISIR
}
//if a Pending ISIR exists, then return it
def pendingISIR = awardYearISIRs.find{it.isirStatus == "Pending"}
if (pendingISIR != null) {
 return pendingISIR
}
//if no ISIRs exist in the Award Year that fit the criteria above, then no ISIR will be returned
return null
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# ISIR C-Code Auto Clearing

Name: isirCCodeAutoCodeClearingLogic

Purpose: Determine if an ISIR c-code is auto cleared.

In Baseline Config: Yes

Return Type: Boolean

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
```

```
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "isirCCodeAutoCodeClearingLogic"
Primary Function: Determine if an ISIR c-code is auto cleared.
Expected Output Type: Boolean
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
AGGREGATES_PERKINS_FLAGS
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
throwsDocumentIsFaUnacceptableExceptionParam
Purpose: This binding can be set within a Groovy script to control whether a subsequent call to
IReceivedDoc.isAcceptable() or IReceivedDocDocumentAPI.isAcceptable() triggers an exception or not when a
document is unnaceptable. The default value is true.
Type: Boolean
Available Methods:
No methods available
GRANT_PAYMENT
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
LOAN_DETAIL
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
OVERPAYMENT
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
allAwardYearReceivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
```

```
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
```

```
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
```

```
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
```

```
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
```

```
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isirCorrection
Purpose: ISIR correction information.
Type: IIsirCorrectionRecordAPI
Available Methods:
* getAwardYearIndicator
 - Returns: String
 - Function: No description available'
* getEdeFederalSchoolCode
 - Returns: String
 - Function: No description available'
* getEmailAddress
 - Returns: String
 - Function: No description available'
* getFieldCorrections
 - Returns: Map<String,String>
 - Function: No description available'
* getHighSchoolName
 - Returns: String
 - Function: No description available'
* getOriginalId
 - Returns: String
 - Function: No description available'
* getParentsEmailAddress
 - Returns: String
 - Function: No description available'
* getStudentSsn
 - Returns: ISocialSecurityNumber
 - Function: No description available'
* getTransactionDataSourceTypeCode
 - Returns: String
 - Function: No description available'
* getTransactionNumber
 - Returns: String
 - Function: No description available'
* getTransactionReceiptDate
```

```
 - Returns: LocalDate
 - Function: No description available'
* hasIsirCorrection
 - Returns: boolean
 - Function: No description available'
isirRecord
Purpose: The ISIR being evaluated.
Type: IIsirRecordAPI
Available Methods:
* getActiveIsirForAwardYear
 - Returns: Boolean
 - Function: Gets the active ISIR for award year status.'
* getAllFields
 - Returns: @NonNull Map<IsirField<?>,IIsirFieldValue<?>>
 - Function: Retrieves all field values (WARNING: can only be used with 2025+ ISIRs).'
* getAssumptions
 - Returns: List<IAssumptionAPI>
 - Function: Get the assumptions.'
* getAwardYear
 - Returns: String
 - Function: Gets the award year.'
* getCommentCodes
 - Returns: List<ICommentCodeAPI>
 - Function: Get the comment codes.'
* getCpsTransNumber
 - Returns: String
 - Function: Gets the CPS transaction number.'
* getCurrentSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's current SSN.'
* getDiscrepancies
 - Returns: List<IDiscrepancyAPI>
 - Function: Get the discrepancies.'
* getField
 - Returns: <T> @NonNull IIsirFieldValue<T>
 - Function: Retrieves the value of a given field (WARNING: can only be used with 2025+ ISIRs).'
* getFirstName
 - Returns: String
 - Function: Gets the student's first name.'
* getHighlightFlags
 - Returns: List<IHighlightFlagAPI>
 - Function: Get the highlight flags.'
* getIsirFieldValue
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValueOrNull
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValues
 - Returns: Map<String,String>
 - Function: Get a map containing all ISIR fields (WARNING: This method can only be used up to 2024 award
 year).'
* getIsirIdOriginalNameId
 - Returns: String
 - Function: Gets the student's original name ID.'
* getIsirIdSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's original SSN.'
* getIsirIdYearIndicator
 - Returns: String
 - Function: Gets the ISIR's year indicator.'
* getIsirStatus
 - Returns: String
 - Function: Gets the ISIR status.'
* getIsirStatusBeforeCorrection
 - Returns: String
```

```
 - Function: When ISIR Status is Corrected returns the previous ISIR Status.'
* getLastName
 - Returns: String
 - Function: Gets the student's last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the student's middle name.'
* getReceivedDate
 - Returns: Date
 - Function: Gets the system receipt date of the ISIR.'
* getRejectCodes
 - Returns: List<IRejectCodeAPI>
 - Function: Get the reject codes.'
* getSchoolCode1
 - Returns: String
 - Function: Gets the school code 1.'
* getSchoolCode10
 - Returns: String
 - Function: Gets the school code 10.'
* getSchoolCode2
 - Returns: String
 - Function: Gets the school code 2.'
* getSchoolCode3
 - Returns: String
 - Function: Gets the school code 3.'
* getSchoolCode4
 - Returns: String
 - Function: Gets the school code 4.'
* getSchoolCode5
 - Returns: String
 - Function: Gets the school code 5.'
* getSchoolCode6
 - Returns: String
 - Function: Gets the school code 6.'
* getSchoolCode7
 - Returns: String
 - Function: Gets the school code 7.'
* getSchoolCode8
 - Returns: String
 - Function: Gets the school code 8.'
* getSchoolCode9
 - Returns: String
 - Function: Gets the school code 9.'
* isCompatibleWithPackaging
 - Returns: boolean
 - Function: Checks if this ISIR satisfies minimal acceptance criteria for automatic Financial
Aid Package Calculation.'
* isUnverified
 - Returns: boolean
 - Function: Gets the ISIR verification status.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
```

```
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
nsldsGuarantorServicers
Purpose: List of all guarantors and servicers.
Type: IGuarantorServicersListAPI
Available Methods:
* getAwardYear
 - Returns: String
 - Function: No description available'
* getGuarantorServicerName
 - Returns: String
 - Function: No description available'
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
```

```
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
```

```
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
```

```
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Active documents for the student for the award year of the ISIR being evaluated.
Type: IReceivedDocs
Available Methods:
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and Student, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and owners, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code, owners, and subtype, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and Student, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and owners, if it exists.'
* getValidDocCodes
 - Returns: Set<String>
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: Return true if doc exists with the right code and owners'
* noDocs
 - Returns: boolean
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
recentFahRecords
Purpose: All Financial Aid History records associated with the student.
Type: IRecentFahRecordsAPI
Available Methods:
* hasRecordsOfType
 - Returns: boolean
 - Function: No description available'
* overlapsLoan
 - Returns: boolean
 - Function: No description available'
* getRecords
 - Returns: List<IFahRecordAPI>
 - Function: No description available'
* getRecordsByType
```

```
 - Returns: List<IFahRecordAPI>
 - Function: No description available'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
```

```
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentFollowUp
Purpose: Binding that can be populated by the script to flag the c-code as "Student Follow-Up".
Type: IStudentFollowUp
Available Methods:
* getStudentFollowUpReason
 - Returns: String
 - Function: No description available'
* isStudentFollowUpRequired
 - Returns: boolean
 - Function: No description available'
* setStudentFollowUpReason
 - Returns: void
 - Function: No description available'
* setStudentFollowUpRequired
 - Returns: void
 - Function: No description available'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
```

```
    // Error Handling: [how errors are managed]
    // Financial Impact: [effect on aid calculations]
    // If N/A, explicitly state: "CONDITIONALS: N/A"
   2. BOUNDARY CHECKS Section:
    Format:
    // BOUNDARY CHECKS:
    // Check [number]: [specific limit being enforced]
    // Source: [where boundary values come from]
    // Variables: [variables and methods involved]
    // Validation:
    // - [how boundaries are checked]
    // - [validation method details]
    // Edge Cases:
    // - [specific edge cases handled]
    // - [how each is managed]
    // Error Handling: [boundary violation management]
    // Impact: [effect of boundary violations]
    // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
   3. ARITHMETIC Section:
    Format:
    // ARITHMETIC:
    // Operation [number]: [name of calculation]
    // Formula: [complete mathematical expression]
    // Variables:
    // - [each variable and its purpose]
    // - [data types and units]
    // Steps:
    // 1. [detailed step-by-step breakdown]
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
   ADDITIONAL REQUIREMENTS:
   1. Complex Logic Sections:
    - Must break down multi-step operations
    - Explain interactions between components
    - Detail data flow and transformations
   2. Error Handling:
    - Document all error conditions
    - Explain recovery procedures
    - Detail logging and reporting
   3. API Interactions:
    - Document all platform method calls
    - Explain data transformations
    - Detail error handling
   4. Financial Aid Impact:
    - Explain effects on aid calculations
    - Document regulatory compliance
    - Detail student eligibility impact
   Example Section With All Aspects:
   // CONDITIONALS:
   // Check 1: Payment Period Status Validation
   // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
   // Purpose: Ensure only valid, active payment periods are included
   // Logic:
   // - Retrieve all loan payment periods
   // - Filter based on status (!= "CANCELED")
   // - Compare start dates with reference date
   // Success Path: Period included in calculations
   // Failure Path: Period excluded from totals
   // Business Rules: Canceled periods must not affect grade level
   // Error Handling: Invalid periods logged and skipped
   // Financial Impact: Affects available loan amounts
```

```
    //
    // BOUNDARY CHECKS:
    // Check 1: Payment Period Date Validation
    // Source: Academic Calendar Configuration
    // Variables: period.getStartDate(), referenceDate
    // Validation:
    // - Compare period start date with reference date
    // - Ensure dates are not null
    // Edge Cases:
    // - Null dates are excluded
    // - Future dates are filtered
    // Error Handling: Invalid dates logged and skipped
    // Impact: Prevents future periods from affecting current calculation
    //
    // ARITHMETIC:
    // Operation 1: Payment Period Units Summation
    // Formula: sum(validPeriods.map(period => period.getUnits()))
    // Variables:
    // - period.getUnits(): Decimal, represents credit hours
    // - validPeriods: Array of filtered payment periods
    // Steps:
    // 1. Filter valid payment periods
    // 2. Extract units from each period
    // 3. Sum all units using reduce
    // Business Context: Calculate total completed units for grade level
    // Result: Total units affecting grade level
    // Financial Impact: Determines loan eligibility amounts
    // Limitations: Cannot exceed program maximum units
    ## Groovy Script for Analysis
    ```groovy
    //file:noinspection UnnecessaryQualifiedReference
    @groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.IsirCCodeAutoCodeClearingLogicScript)
    package CHANGE_ME // this should be updated to your actual package name
    import com.oracle.sfp.scripting.api.*
    import com.oracle.sfp.scripting.api.util.*
    studentFollowUp.studentFollowUpRequired = true;
    return false;
    ```
    ## COMMENT GENERATION REQUIREMENTS:
    1. Add comprehensive comments following the exact format specified
    2. Every logical section must include all three aspects:
     - CONDITIONALS
     - BOUNDARY CHECKS
     - ARITHMETIC
    3. Mark any non-applicable aspect as "N/A" explicitly
    4. Ensure line-by-line explanation of complex logic
    5. Detail all financial aid implications
    DO NOT modify the code itself. Add comments only.
    REMEMBER:
    - Comments must be exhaustively detailed
    - Every logical section needs all three aspects
    - Financial aid impact must be clear
    - Business rules must be explicit
    Return ONLY the fully commented Groovy script. No other text or explanations.
```

# ISIR Assumption Auto Code Clearing

Name: isirAssumptionAutoCodeClearingLogic

Purpose: Determine if an ISIR assumption is auto cleared.

In Baseline Config: No

Return Type: Boolean

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "isirAssumptionAutoCodeClearingLogic"
Primary Function: Determine if an ISIR assumption is auto cleared.
Expected Output Type: Boolean
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
AGGREGATES_PERKINS_FLAGS
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
GRANT_PAYMENT
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
LOAN_DETAIL
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
OVERPAYMENT
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
allAwardYearReceivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
```

```
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
```

```
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
```

```
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
```

```
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
```

```
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isirCorrection
Purpose: ISIR correction information.
Type: IIsirCorrectionRecordAPI
Available Methods:
* getAwardYearIndicator
 - Returns: String
 - Function: No description available'
* getEdeFederalSchoolCode
 - Returns: String
 - Function: No description available'
* getEmailAddress
 - Returns: String
 - Function: No description available'
* getFieldCorrections
 - Returns: Map<String,String>
 - Function: No description available'
* getHighSchoolName
 - Returns: String
 - Function: No description available'
* getOriginalId
 - Returns: String
 - Function: No description available'
* getParentsEmailAddress
 - Returns: String
 - Function: No description available'
* getStudentSsn
 - Returns: ISocialSecurityNumber
 - Function: No description available'
* getTransactionDataSourceTypeCode
 - Returns: String
 - Function: No description available'
* getTransactionNumber
 - Returns: String
 - Function: No description available'
* getTransactionReceiptDate
 - Returns: LocalDate
 - Function: No description available'
* hasIsirCorrection
```

```
 - Returns: boolean
 - Function: No description available'
isirRecord
Purpose: The ISIR being evaluated.
Type: IIsirRecordAPI
Available Methods:
* getActiveIsirForAwardYear
 - Returns: Boolean
 - Function: Gets the active ISIR for award year status.'
* getAllFields
 - Returns: @NonNull Map<IsirField<?>,IIsirFieldValue<?>>
 - Function: Retrieves all field values (WARNING: can only be used with 2025+ ISIRs).'
* getAssumptions
 - Returns: List<IAssumptionAPI>
 - Function: Get the assumptions.'
* getAwardYear
 - Returns: String
 - Function: Gets the award year.'
* getCommentCodes
 - Returns: List<ICommentCodeAPI>
 - Function: Get the comment codes.'
* getCpsTransNumber
 - Returns: String
 - Function: Gets the CPS transaction number.'
* getCurrentSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's current SSN.'
* getDiscrepancies
 - Returns: List<IDiscrepancyAPI>
 - Function: Get the discrepancies.'
* getField
 - Returns: <T> @NonNull IIsirFieldValue<T>
 - Function: Retrieves the value of a given field (WARNING: can only be used with 2025+ ISIRs).'
* getFirstName
 - Returns: String
 - Function: Gets the student's first name.'
* getHighlightFlags
 - Returns: List<IHighlightFlagAPI>
 - Function: Get the highlight flags.'
* getIsirFieldValue
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValueOrNull
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValues
 - Returns: Map<String,String>
 - Function: Get a map containing all ISIR fields (WARNING: This method can only be used up to 2024 award
 year).'
* getIsirIdOriginalNameId
 - Returns: String
 - Function: Gets the student's original name ID.'
* getIsirIdSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's original SSN.'
* getIsirIdYearIndicator
 - Returns: String
 - Function: Gets the ISIR's year indicator.'
* getIsirStatus
 - Returns: String
 - Function: Gets the ISIR status.'
* getIsirStatusBeforeCorrection
 - Returns: String
 - Function: When ISIR Status is Corrected returns the previous ISIR Status.'
* getLastName
 - Returns: String
```

```
 - Function: Gets the student's last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the student's middle name.'
* getReceivedDate
 - Returns: Date
 - Function: Gets the system receipt date of the ISIR.'
* getRejectCodes
 - Returns: List<IRejectCodeAPI>
 - Function: Get the reject codes.'
* getSchoolCode1
 - Returns: String
 - Function: Gets the school code 1.'
* getSchoolCode10
 - Returns: String
 - Function: Gets the school code 10.'
* getSchoolCode2
 - Returns: String
 - Function: Gets the school code 2.'
* getSchoolCode3
 - Returns: String
 - Function: Gets the school code 3.'
* getSchoolCode4
 - Returns: String
 - Function: Gets the school code 4.'
* getSchoolCode5
 - Returns: String
 - Function: Gets the school code 5.'
* getSchoolCode6
 - Returns: String
 - Function: Gets the school code 6.'
* getSchoolCode7
 - Returns: String
 - Function: Gets the school code 7.'
* getSchoolCode8
 - Returns: String
 - Function: Gets the school code 8.'
* getSchoolCode9
 - Returns: String
 - Function: Gets the school code 9.'
* isCompatibleWithPackaging
 - Returns: boolean
 - Function: Checks if this ISIR satisfies minimal acceptance criteria for automatic Financial
Aid Package Calculation.'
* isUnverified
 - Returns: boolean
 - Function: Gets the ISIR verification status.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
```

```
* warn
 - Returns: void
 - Function: No description available'
nsldsGuarantorServicers
Purpose: List of all guarantors and servicers.
Type: IGuarantorServicersListAPI
Available Methods:
* getAwardYear
 - Returns: String
 - Function: No description available'
* getGuarantorServicerName
 - Returns: String
 - Function: No description available'
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
```

```
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
```

```
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
```

```
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Active documents for the student for the award year of the ISIR being evaluated.
Type: IReceivedDocs
Available Methods:
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and Student, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and owners, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code, owners, and subtype, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and Student, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and owners, if it exists.'
* getValidDocCodes
 - Returns: Set<String>
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: Return true if doc exists with the right code and owners'
* noDocs
 - Returns: boolean
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
recentFahRecords
Purpose: All Financial Aid History records associated with the student.
Type: IRecentFahRecordsAPI
Available Methods:
* hasRecordsOfType
 - Returns: boolean
 - Function: No description available'
* overlapsLoan
 - Returns: boolean
 - Function: No description available'
* getRecords
 - Returns: List<IFahRecordAPI>
 - Function: No description available'
* getRecordsByType
 - Returns: List<IFahRecordAPI>
 - Function: No description available'
student
```

```
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
```

```
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentFollowUp
Purpose: Binding that can be populated by the script to flag the assumption as "Student Follow-Up".
Type: IStudentFollowUp
Available Methods:
* getStudentFollowUpReason
 - Returns: String
 - Function: No description available'
* isStudentFollowUpRequired
 - Returns: boolean
 - Function: No description available'
* setStudentFollowUpReason
 - Returns: void
 - Function: No description available'
* setStudentFollowUpRequired
 - Returns: void
 - Function: No description available'
throwsDocumentIsFaUnacceptableExceptionParam
Purpose: This binding can be set within a Groovy script to control whether a subsequent call to
IReceivedDoc.isAcceptable() or IReceivedDocDocumentAPI.isAcceptable() triggers an exception or not when a
document is unnaceptable. The default value is true.
Type: Boolean
Available Methods:
No methods available
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
```

```
    // - [any transformations applied]
    // Success Path: [what happens when condition is true]
    // Failure Path: [what happens when condition is false]
    // Business Rules: [specific rules being enforced]
    // Error Handling: [how errors are managed]
    // Financial Impact: [effect on aid calculations]
    // If N/A, explicitly state: "CONDITIONALS: N/A"
   2. BOUNDARY CHECKS Section:
    Format:
    // BOUNDARY CHECKS:
    // Check [number]: [specific limit being enforced]
    // Source: [where boundary values come from]
    // Variables: [variables and methods involved]
    // Validation:
    // - [how boundaries are checked]
    // - [validation method details]
    // Edge Cases:
    // - [specific edge cases handled]
    // - [how each is managed]
    // Error Handling: [boundary violation management]
    // Impact: [effect of boundary violations]
    // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
   3. ARITHMETIC Section:
    Format:
    // ARITHMETIC:
    // Operation [number]: [name of calculation]
    // Formula: [complete mathematical expression]
    // Variables:
    // - [each variable and its purpose]
    // - [data types and units]
    // Steps:
    // 1. [detailed step-by-step breakdown]
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
   ADDITIONAL REQUIREMENTS:
   1. Complex Logic Sections:
    - Must break down multi-step operations
    - Explain interactions between components
    - Detail data flow and transformations
   2. Error Handling:
    - Document all error conditions
    - Explain recovery procedures
    - Detail logging and reporting
   3. API Interactions:
    - Document all platform method calls
    - Explain data transformations
    - Detail error handling
   4. Financial Aid Impact:
    - Explain effects on aid calculations
    - Document regulatory compliance
    - Detail student eligibility impact
   Example Section With All Aspects:
   // CONDITIONALS:
   // Check 1: Payment Period Status Validation
   // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
   // Purpose: Ensure only valid, active payment periods are included
   // Logic:
   // - Retrieve all loan payment periods
   // - Filter based on status (!= "CANCELED")
   // - Compare start dates with reference date
   // Success Path: Period included in calculations
```

```
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
N/A - There is no baseline config. This is a placeholder.
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# ISIR Highlight Auto Code Clearing

Name: isirHighlightAutoCodeClearingLogic

Purpose: Determine if an ISIR highlight is auto cleared.

In Baseline Config: No

Return Type: Boolean

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
```

```
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "isirHighlightAutoCodeClearingLogic"
Primary Function: Determine if an ISIR highlight is auto cleared.
Expected Output Type: Boolean
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
AGGREGATES_PERKINS_FLAGS
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
GRANT_PAYMENT
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
LOAN_DETAIL
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
OVERPAYMENT
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
allAwardYearReceivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
```

```
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
```

```
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
```

```
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
```

```
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
```

```
    - Returns: Integer
    - Function: Parses a number represented in ISIR format to Integer.'
  * getIsirDate
    - Returns: Date
    - Function: No description available'
  * getIsirDateFormatStr
    - Returns: String
    - Function: No description available'
  * getSignedIsirNumberFromInteger
    - Returns: String
    - Function: No description available'
  * isirRecordsDifferForListedFields
    - Returns: boolean
    - Function: No description available'
  * mapRanges
    - Returns: IRangeMapStarterAPI
    - Function: No description available'
  * newAwardYearPeriod
    - Returns: IAwardYearAPI
    - Function: Returns the Award Year having a given end year.'
  * newPeriod
    - Returns: IPeriod
    - Function: Returns the IPeriod for one single day.'
  * newPeriod
    - Returns: IPeriod
    - Function: Returns a IPeriod given its start and end dates.'
isirCorrection
Purpose: ISIR correction information.
Type: IIsirCorrectionRecordAPI
Available Methods:
  * getAwardYearIndicator
    - Returns: String
    - Function: No description available'
  * getEdeFederalSchoolCode
    - Returns: String
    - Function: No description available'
  * getEmailAddress
    - Returns: String
    - Function: No description available'
  * getFieldCorrections
    - Returns: Map<String,String>
    - Function: No description available'
  * getHighSchoolName
    - Returns: String
    - Function: No description available'
  * getOriginalId
    - Returns: String
    - Function: No description available'
  * getParentsEmailAddress
    - Returns: String
    - Function: No description available'
  * getStudentSsn
    - Returns: ISocialSecurityNumber
    - Function: No description available'
  * getTransactionDataSourceTypeCode
    - Returns: String
    - Function: No description available'
  * getTransactionNumber
    - Returns: String
    - Function: No description available'
  * getTransactionReceiptDate
    - Returns: LocalDate
    - Function: No description available'
  * hasIsirCorrection
    - Returns: boolean
    - Function: No description available'
```

```
isirRecord
Purpose: The ISIR being evaluated.
Type: IIsirRecordAPI
Available Methods:
* getActiveIsirForAwardYear
 - Returns: Boolean
 - Function: Gets the active ISIR for award year status.'
* getAllFields
 - Returns: @NonNull Map<IsirField<?>,IIsirFieldValue<?>>
 - Function: Retrieves all field values (WARNING: can only be used with 2025+ ISIRs).'
* getAssumptions
 - Returns: List<IAssumptionAPI>
 - Function: Get the assumptions.'
* getAwardYear
 - Returns: String
 - Function: Gets the award year.'
* getCommentCodes
 - Returns: List<ICommentCodeAPI>
 - Function: Get the comment codes.'
* getCpsTransNumber
 - Returns: String
 - Function: Gets the CPS transaction number.'
* getCurrentSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's current SSN.'
* getDiscrepancies
 - Returns: List<IDiscrepancyAPI>
 - Function: Get the discrepancies.'
* getField
 - Returns: <T> @NonNull IIsirFieldValue<T>
 - Function: Retrieves the value of a given field (WARNING: can only be used with 2025+ ISIRs).'
* getFirstName
 - Returns: String
 - Function: Gets the student's first name.'
* getHighlightFlags
 - Returns: List<IHighlightFlagAPI>
 - Function: Get the highlight flags.'
* getIsirFieldValue
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValueOrNull
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValues
 - Returns: Map<String,String>
 - Function: Get a map containing all ISIR fields (WARNING: This method can only be used up to 2024 award
 year).'
* getIsirIdOriginalNameId
 - Returns: String
 - Function: Gets the student's original name ID.'
* getIsirIdSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's original SSN.'
* getIsirIdYearIndicator
 - Returns: String
 - Function: Gets the ISIR's year indicator.'
* getIsirStatus
 - Returns: String
 - Function: Gets the ISIR status.'
* getIsirStatusBeforeCorrection
 - Returns: String
 - Function: When ISIR Status is Corrected returns the previous ISIR Status.'
* getLastName
 - Returns: String
 - Function: Gets the student's last name.'
* getMiddleName
```

```
 - Returns: String
 - Function: Gets the student's middle name.'
* getReceivedDate
 - Returns: Date
 - Function: Gets the system receipt date of the ISIR.'
* getRejectCodes
 - Returns: List<IRejectCodeAPI>
 - Function: Get the reject codes.'
* getSchoolCode1
 - Returns: String
 - Function: Gets the school code 1.'
* getSchoolCode10
 - Returns: String
 - Function: Gets the school code 10.'
* getSchoolCode2
 - Returns: String
 - Function: Gets the school code 2.'
* getSchoolCode3
 - Returns: String
 - Function: Gets the school code 3.'
* getSchoolCode4
 - Returns: String
 - Function: Gets the school code 4.'
* getSchoolCode5
 - Returns: String
 - Function: Gets the school code 5.'
* getSchoolCode6
 - Returns: String
 - Function: Gets the school code 6.'
* getSchoolCode7
 - Returns: String
 - Function: Gets the school code 7.'
* getSchoolCode8
 - Returns: String
 - Function: Gets the school code 8.'
* getSchoolCode9
 - Returns: String
 - Function: Gets the school code 9.'
* isCompatibleWithPackaging
 - Returns: boolean
 - Function: Checks if this ISIR satisfies minimal acceptance criteria for automatic Financial
Aid Package Calculation.'
* isUnverified
 - Returns: boolean
 - Function: Gets the ISIR verification status.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
```

```
 - Function: No description available'
nsldsGuarantorServicers
Purpose: List of all guarantors and servicers.
Type: IGuarantorServicersListAPI
Available Methods:
* getAwardYear
 - Returns: String
 - Function: No description available'
* getGuarantorServicerName
 - Returns: String
 - Function: No description available'
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
```

```
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
```

```
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
```

```
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Active documents for the student for the award year of the ISIR being evaluated.
Type: IReceivedDocs
Available Methods:
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and Student, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and owners, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code, owners, and subtype, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and Student, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and owners, if it exists.'
* getValidDocCodes
 - Returns: Set<String>
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: Return true if doc exists with the right code and owners'
* noDocs
 - Returns: boolean
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
recentFahRecords
Purpose: All Financial Aid History records associated with the student.
Type: IRecentFahRecordsAPI
Available Methods:
* hasRecordsOfType
 - Returns: boolean
 - Function: No description available'
* overlapsLoan
 - Returns: boolean
 - Function: No description available'
* getRecords
 - Returns: List<IFahRecordAPI>
 - Function: No description available'
* getRecordsByType
 - Returns: List<IFahRecordAPI>
 - Function: No description available'
student
Purpose: Student information.
Type: IFasStudentAPI
```

```
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
```

```
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentFollowUp
Purpose: Binding that can be populated by the script to flag the highlight as "Student Follow-Up".
Type: IStudentFollowUp
Available Methods:
* getStudentFollowUpReason
 - Returns: String
 - Function: No description available'
* isStudentFollowUpRequired
 - Returns: boolean
 - Function: No description available'
* setStudentFollowUpReason
 - Returns: void
 - Function: No description available'
* setStudentFollowUpRequired
 - Returns: void
 - Function: No description available'
throwsDocumentIsFaUnacceptableExceptionParam
Purpose: This binding can be set within a Groovy script to control whether a subsequent call to
IReceivedDoc.isAcceptable() or IReceivedDocDocumentAPI.isAcceptable() triggers an exception or not when a
document is unnaceptable. The default value is true.
Type: Boolean
Available Methods:
No methods available
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
```

```
    // Failure Path: [what happens when condition is false]
    // Business Rules: [specific rules being enforced]
    // Error Handling: [how errors are managed]
    // Financial Impact: [effect on aid calculations]
    // If N/A, explicitly state: "CONDITIONALS: N/A"
   2. BOUNDARY CHECKS Section:
    Format:
    // BOUNDARY CHECKS:
    // Check [number]: [specific limit being enforced]
    // Source: [where boundary values come from]
    // Variables: [variables and methods involved]
    // Validation:
    // - [how boundaries are checked]
    // - [validation method details]
    // Edge Cases:
    // - [specific edge cases handled]
    // - [how each is managed]
    // Error Handling: [boundary violation management]
    // Impact: [effect of boundary violations]
    // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
   3. ARITHMETIC Section:
    Format:
    // ARITHMETIC:
    // Operation [number]: [name of calculation]
    // Formula: [complete mathematical expression]
    // Variables:
    // - [each variable and its purpose]
    // - [data types and units]
    // Steps:
    // 1. [detailed step-by-step breakdown]
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
   ADDITIONAL REQUIREMENTS:
   1. Complex Logic Sections:
    - Must break down multi-step operations
    - Explain interactions between components
    - Detail data flow and transformations
   2. Error Handling:
    - Document all error conditions
    - Explain recovery procedures
    - Detail logging and reporting
   3. API Interactions:
    - Document all platform method calls
    - Explain data transformations
    - Detail error handling
   4. Financial Aid Impact:
    - Explain effects on aid calculations
    - Document regulatory compliance
    - Detail student eligibility impact
   Example Section With All Aspects:
   // CONDITIONALS:
   // Check 1: Payment Period Status Validation
   // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
   // Purpose: Ensure only valid, active payment periods are included
   // Logic:
   // - Retrieve all loan payment periods
   // - Filter based on status (!= "CANCELED")
   // - Compare start dates with reference date
   // Success Path: Period included in calculations
   // Failure Path: Period excluded from totals
   // Business Rules: Canceled periods must not affect grade level
```

```
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
N/A - There is no baseline config. This is a placeholder.
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# ISIR Reject Code Auto Code Clearing

Name: isirRejectCodeAutoCodeClearingLogic

Purpose: Determine if an ISIR reject code is auto cleared.

In Baseline Config: No

Return Type: Boolean

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
```

```
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "isirRejectCodeAutoCodeClearingLogic"
Primary Function: Determine if an ISIR reject code is auto cleared.
Expected Output Type: Boolean
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
AGGREGATES_PERKINS_FLAGS
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
GRANT_PAYMENT
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
LOAN_DETAIL
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
OVERPAYMENT
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
allAwardYearReceivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
```

```
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
```

```
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
```

```
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
```

```
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
```

```
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isirCorrection
Purpose: ISIR correction information.
Type: IIsirCorrectionRecordAPI
Available Methods:
* getAwardYearIndicator
 - Returns: String
 - Function: No description available'
* getEdeFederalSchoolCode
 - Returns: String
 - Function: No description available'
* getEmailAddress
 - Returns: String
 - Function: No description available'
* getFieldCorrections
 - Returns: Map<String,String>
 - Function: No description available'
* getHighSchoolName
 - Returns: String
 - Function: No description available'
* getOriginalId
 - Returns: String
 - Function: No description available'
* getParentsEmailAddress
 - Returns: String
 - Function: No description available'
* getStudentSsn
 - Returns: ISocialSecurityNumber
 - Function: No description available'
* getTransactionDataSourceTypeCode
 - Returns: String
 - Function: No description available'
* getTransactionNumber
 - Returns: String
 - Function: No description available'
* getTransactionReceiptDate
 - Returns: LocalDate
 - Function: No description available'
* hasIsirCorrection
 - Returns: boolean
 - Function: No description available'
isirRecord
Purpose: The ISIR being evaluated.
Type: IIsirRecordAPI
```

```
Available Methods:
* getActiveIsirForAwardYear
 - Returns: Boolean
 - Function: Gets the active ISIR for award year status.'
* getAllFields
 - Returns: @NonNull Map<IsirField<?>,IIsirFieldValue<?>>
 - Function: Retrieves all field values (WARNING: can only be used with 2025+ ISIRs).'
* getAssumptions
 - Returns: List<IAssumptionAPI>
 - Function: Get the assumptions.'
* getAwardYear
 - Returns: String
 - Function: Gets the award year.'
* getCommentCodes
 - Returns: List<ICommentCodeAPI>
 - Function: Get the comment codes.'
* getCpsTransNumber
 - Returns: String
 - Function: Gets the CPS transaction number.'
* getCurrentSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's current SSN.'
* getDiscrepancies
 - Returns: List<IDiscrepancyAPI>
 - Function: Get the discrepancies.'
* getField
 - Returns: <T> @NonNull IIsirFieldValue<T>
 - Function: Retrieves the value of a given field (WARNING: can only be used with 2025+ ISIRs).'
* getFirstName
 - Returns: String
 - Function: Gets the student's first name.'
* getHighlightFlags
 - Returns: List<IHighlightFlagAPI>
 - Function: Get the highlight flags.'
* getIsirFieldValue
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValueOrNull
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValues
 - Returns: Map<String,String>
 - Function: Get a map containing all ISIR fields (WARNING: This method can only be used up to 2024 award
 year).'
* getIsirIdOriginalNameId
 - Returns: String
 - Function: Gets the student's original name ID.'
* getIsirIdSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's original SSN.'
* getIsirIdYearIndicator
 - Returns: String
 - Function: Gets the ISIR's year indicator.'
* getIsirStatus
 - Returns: String
 - Function: Gets the ISIR status.'
* getIsirStatusBeforeCorrection
 - Returns: String
 - Function: When ISIR Status is Corrected returns the previous ISIR Status.'
* getLastName
 - Returns: String
 - Function: Gets the student's last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the student's middle name.'
* getReceivedDate
```

```
 - Returns: Date
 - Function: Gets the system receipt date of the ISIR.'
* getRejectCodes
 - Returns: List<IRejectCodeAPI>
 - Function: Get the reject codes.'
* getSchoolCode1
 - Returns: String
 - Function: Gets the school code 1.'
* getSchoolCode10
 - Returns: String
 - Function: Gets the school code 10.'
* getSchoolCode2
 - Returns: String
 - Function: Gets the school code 2.'
* getSchoolCode3
 - Returns: String
 - Function: Gets the school code 3.'
* getSchoolCode4
 - Returns: String
 - Function: Gets the school code 4.'
* getSchoolCode5
 - Returns: String
 - Function: Gets the school code 5.'
* getSchoolCode6
 - Returns: String
 - Function: Gets the school code 6.'
* getSchoolCode7
 - Returns: String
 - Function: Gets the school code 7.'
* getSchoolCode8
 - Returns: String
 - Function: Gets the school code 8.'
* getSchoolCode9
 - Returns: String
 - Function: Gets the school code 9.'
* isCompatibleWithPackaging
 - Returns: boolean
 - Function: Checks if this ISIR satisfies minimal acceptance criteria for automatic Financial
Aid Package Calculation.'
* isUnverified
 - Returns: boolean
 - Function: Gets the ISIR verification status.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
nsldsGuarantorServicers
Purpose: List of all guarantors and servicers.
```

```
Type: IGuarantorServicersListAPI
Available Methods:
* getAwardYear
 - Returns: String
 - Function: No description available'
* getGuarantorServicerName
 - Returns: String
 - Function: No description available'
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
```

```
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
```

```
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
```

```
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Active documents for the student for the award year of the ISIR being evaluated.
Type: IReceivedDocs
Available Methods:
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and Student, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and owners, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code, owners, and subtype, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and Student, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and owners, if it exists.'
* getValidDocCodes
 - Returns: Set<String>
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: Return true if doc exists with the right code and owners'
* noDocs
 - Returns: boolean
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
recentFahRecords
Purpose: All Financial Aid History records associated with the student.
Type: IRecentFahRecordsAPI
Available Methods:
* hasRecordsOfType
 - Returns: boolean
 - Function: No description available'
* overlapsLoan
 - Returns: boolean
 - Function: No description available'
* getRecords
 - Returns: List<IFahRecordAPI>
 - Function: No description available'
* getRecordsByType
 - Returns: List<IFahRecordAPI>
 - Function: No description available'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
```

```
   - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
  * getBahIndicator
   - Returns: Boolean
   - Function: Gets the school employee dependet status.'
  * getCampusCode
   - Returns: String
   - Function: Gets the campus code.'
  * getCompletionDate
   - Returns: Date
   - Function: Gets the completion date.'
  * getDateOfBirth
   - Returns: ISecureLocalDate
   - Function: Gets the date of birth.'
  * getDependencyStatus
   - Returns: String
   - Function: Gets the dependency status.'
  * getDeterminationDate
   - Returns: Date
   - Function: Gets the determination date.'
  * getDocuments
   - Returns: Map<String,IReceivedDoc>
   - Function: Gets the student's documents, indexed by document code.'
  * getEmail
   - Returns: String
   - Function: Gets the e-mail.'
  * getEnrollmentStatus
   - Returns: String
   - Function: Gets the enrollment status.'
  * getExternalStudentId
   - Returns: String
   - Function: Gets the external student ID (unique per student).'
  * getFinancialAidRequested
   - Returns: Boolean
   - Function: Gets the financial aid request status.'
  * getFirstName
   - Returns: String
   - Function: Gets the first name.'
  * getGender
   - Returns: String
   - Function: Gets the gender.'
  * getHighSchoolEquivalency
   - Returns: Boolean
   - Function: Gets the high school equivalency status.'
  * getLastName
   - Returns: String
   - Function: Gets the last name.'
  * getMiddleName
   - Returns: String
   - Function: Gets the middle name.'
  * getPersonId
   - Returns: Long
   - Function: Gets the person ID.'
  * getProgramType
   - Returns: String
   - Function: Gets the program type.'
  * getRecentFinancialAidHistoryRecords
   - Returns: IRecentFahRecordsAPI
   - Function: Gets the recent financial aid records.'
  * getSchoolCode
   - Returns: String
   - Function: Gets the school code.'
  * getSchoolEmployee
   - Returns: Boolean
   - Function: Gets the school employee status.'
  * getSchoolEmployeeDependent
   - Returns: Boolean
```

```
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentFollowUp
Purpose: Binding that can be populated by the script to flag the reject code as "Student Follow-Up".
Type: IStudentFollowUp
Available Methods:
* getStudentFollowUpReason
 - Returns: String
 - Function: No description available'
* isStudentFollowUpRequired
 - Returns: boolean
 - Function: No description available'
* setStudentFollowUpReason
 - Returns: void
 - Function: No description available'
* setStudentFollowUpRequired
 - Returns: void
 - Function: No description available'
throwsDocumentIsFaUnacceptableExceptionParam
Purpose: This binding can be set within a Groovy script to control whether a subsequent call to
IReceivedDoc.isAcceptable() or IReceivedDocDocumentAPI.isAcceptable() triggers an exception or not when a
document is unnaceptable. The default value is true.
Type: Boolean
Available Methods:
No methods available
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
```

```
    // Financial Impact: [effect on aid calculations]
    // If N/A, explicitly state: "CONDITIONALS: N/A"
   2. BOUNDARY CHECKS Section:
    Format:
    // BOUNDARY CHECKS:
    // Check [number]: [specific limit being enforced]
    // Source: [where boundary values come from]
    // Variables: [variables and methods involved]
    // Validation:
    // - [how boundaries are checked]
    // - [validation method details]
    // Edge Cases:
    // - [specific edge cases handled]
    // - [how each is managed]
    // Error Handling: [boundary violation management]
    // Impact: [effect of boundary violations]
    // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
   3. ARITHMETIC Section:
    Format:
    // ARITHMETIC:
    // Operation [number]: [name of calculation]
    // Formula: [complete mathematical expression]
    // Variables:
    // - [each variable and its purpose]
    // - [data types and units]
    // Steps:
    // 1. [detailed step-by-step breakdown]
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
   ADDITIONAL REQUIREMENTS:
   1. Complex Logic Sections:
    - Must break down multi-step operations
    - Explain interactions between components
    - Detail data flow and transformations
   2. Error Handling:
    - Document all error conditions
    - Explain recovery procedures
    - Detail logging and reporting
   3. API Interactions:
    - Document all platform method calls
    - Explain data transformations
    - Detail error handling
   4. Financial Aid Impact:
    - Explain effects on aid calculations
    - Document regulatory compliance
    - Detail student eligibility impact
   Example Section With All Aspects:
   // CONDITIONALS:
   // Check 1: Payment Period Status Validation
   // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
   // Purpose: Ensure only valid, active payment periods are included
   // Logic:
   // - Retrieve all loan payment periods
   // - Filter based on status (!= "CANCELED")
   // - Compare start dates with reference date
   // Success Path: Period included in calculations
   // Failure Path: Period excluded from totals
   // Business Rules: Canceled periods must not affect grade level
   // Error Handling: Invalid periods logged and skipped
   // Financial Impact: Affects available loan amounts
   //
```

```
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
N/A - There is no baseline config. This is a placeholder.
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# ISIR Verification Clearance Rules

Name: isirVerificationClearanceRules

Purpose: Determine if an ISIR verification is auto cleared.

In Baseline Config: Yes

Return Type: Boolean

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
```

```
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "isirVerificationClearanceRules"
Primary Function: Determine if an ISIR verification is auto cleared.
Expected Output Type: Boolean
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
AGGREGATES_PERKINS_FLAGS
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
GRANT_PAYMENT
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
LOAN_DETAIL
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
OVERPAYMENT
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
allAwardYearReceivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
```

```
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
```

```
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
```

```
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
```

```
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
```

```
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isirRecord
Purpose: The ISIR being evaluated.
Type: IIsirRecordAPI
Available Methods:
* getActiveIsirForAwardYear
 - Returns: Boolean
 - Function: Gets the active ISIR for award year status.'
* getAllFields
 - Returns: @NonNull Map<IsirField<?>,IIsirFieldValue<?>>
 - Function: Retrieves all field values (WARNING: can only be used with 2025+ ISIRs).'
* getAssumptions
 - Returns: List<IAssumptionAPI>
 - Function: Get the assumptions.'
* getAwardYear
 - Returns: String
 - Function: Gets the award year.'
* getCommentCodes
 - Returns: List<ICommentCodeAPI>
 - Function: Get the comment codes.'
* getCpsTransNumber
 - Returns: String
 - Function: Gets the CPS transaction number.'
* getCurrentSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's current SSN.'
* getDiscrepancies
 - Returns: List<IDiscrepancyAPI>
 - Function: Get the discrepancies.'
* getField
 - Returns: <T> @NonNull IIsirFieldValue<T>
 - Function: Retrieves the value of a given field (WARNING: can only be used with 2025+ ISIRs).'
* getFirstName
 - Returns: String
 - Function: Gets the student's first name.'
* getHighlightFlags
 - Returns: List<IHighlightFlagAPI>
 - Function: Get the highlight flags.'
* getIsirFieldValue
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValueOrNull
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValues
 - Returns: Map<String,String>
```

```
 - Function: Get a map containing all ISIR fields (WARNING: This method can only be used up to 2024 award
 year).'
* getIsirIdOriginalNameId
 - Returns: String
 - Function: Gets the student's original name ID.'
* getIsirIdSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's original SSN.'
* getIsirIdYearIndicator
 - Returns: String
 - Function: Gets the ISIR's year indicator.'
* getIsirStatus
 - Returns: String
 - Function: Gets the ISIR status.'
* getIsirStatusBeforeCorrection
 - Returns: String
 - Function: When ISIR Status is Corrected returns the previous ISIR Status.'
* getLastName
 - Returns: String
 - Function: Gets the student's last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the student's middle name.'
* getReceivedDate
 - Returns: Date
 - Function: Gets the system receipt date of the ISIR.'
* getRejectCodes
 - Returns: List<IRejectCodeAPI>
 - Function: Get the reject codes.'
* getSchoolCode1
 - Returns: String
 - Function: Gets the school code 1.'
* getSchoolCode10
 - Returns: String
 - Function: Gets the school code 10.'
* getSchoolCode2
 - Returns: String
 - Function: Gets the school code 2.'
* getSchoolCode3
 - Returns: String
 - Function: Gets the school code 3.'
* getSchoolCode4
 - Returns: String
 - Function: Gets the school code 4.'
* getSchoolCode5
 - Returns: String
 - Function: Gets the school code 5.'
* getSchoolCode6
 - Returns: String
 - Function: Gets the school code 6.'
* getSchoolCode7
 - Returns: String
 - Function: Gets the school code 7.'
* getSchoolCode8
 - Returns: String
 - Function: Gets the school code 8.'
* getSchoolCode9
 - Returns: String
 - Function: Gets the school code 9.'
* isCompatibleWithPackaging
 - Returns: boolean
 - Function: Checks if this ISIR satisfies minimal acceptance criteria for automatic Financial
Aid Package Calculation.'
* isUnverified
 - Returns: boolean
 - Function: Gets the ISIR verification status.'
```

```
receivedDocuments
Purpose: Active documents for the student for the award year of the ISIR being evaluated.
Type: IReceivedDocs
Available Methods:
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and Student, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and owners, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code, owners, and subtype, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and Student, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and owners, if it exists.'
* getValidDocCodes
 - Returns: Set<String>
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: Return true if doc exists with the right code and owners'
* noDocs
 - Returns: boolean
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
recentFahRecords
Purpose: All Financial Aid History records associated with the student.
Type: IRecentFahRecordsAPI
Available Methods:
* hasRecordsOfType
 - Returns: boolean
 - Function: No description available'
* overlapsLoan
 - Returns: boolean
 - Function: No description available'
* getRecords
 - Returns: List<IFahRecordAPI>
 - Function: No description available'
* getRecordsByType
 - Returns: List<IFahRecordAPI>
 - Function: No description available'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
```

```
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
```

```
throwsDocumentIsFaUnacceptableExceptionParam
Purpose: This binding can be set within a Groovy script to control whether a subsequent call to
IReceivedDoc.isAcceptable() or IReceivedDocDocumentAPI.isAcceptable() triggers an exception or not when a
document is unnaceptable. The default value is true.
Type: Boolean
Available Methods:
No methods available
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
nsldsGuarantorServicers
Purpose: List of all guarantors and servicers.
Type: IGuarantorServicersListAPI
Available Methods:
* getAwardYear
 - Returns: String
 - Function: No description available'
* getGuarantorServicerName
 - Returns: String
 - Function: No description available'
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
```

```
    - Returns: String
    - Function: No description available'
  * getAdmissionStatus
    - Returns: String
    - Function: No description available'
  * getAssessedUnits
    - Returns: BigDecimal
    - Function: No description available'
  * getAydUnits
    - Returns: BigDecimal
    - Function: No description available'
  * getAydWeeks
    - Returns: Integer
    - Function: No description available'
  * getCharges
    - Returns: List<IChargeAPI>
    - Function: No description available'
  * getCode
    - Returns: String
    - Function: No description available'
  * getCollege
    - Returns: String
    - Function: No description available'
  * getCourses
    - Returns: ICoursesAPI
    - Function: No description available'
  * getDateOfDetermination
    - Returns: LocalDate
    - Function: No description available'
  * getDescription
    - Returns: String
    - Function: No description available'
  * getEnrollmentStatus
    - Returns: String
    - Function: No description available'
  * getEnrollmentStatusEffectiveDate
    - Returns: LocalDate
    - Function: No description available'
  * getEnrollmentStatusSubType
    - Returns: String
    - Function: No description available'
  * getExternalProgramId
    - Returns: String
    - Function: No description available'
  * getFaEligibleIndicator
    - Returns: Boolean
    - Function: No description available'
  * getGpa
    - Returns: BigDecimal
    - Function: No description available'
  * getManualSapEvaluationIndicator
    - Returns: Boolean
    - Function: No description available'
  * getModality
    - Returns: String
    - Function: No description available'
  * getOfficialLastDateOfAttendance
    - Returns: LocalDate
    - Function: No description available'
  * getOpeId
    - Returns: String
    - Function: No description available'
  * getPrimaryLocation
    - Returns: String
    - Function: No description available'
  * getProgramLengthMeasurementUnit
```

```
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
```

```
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
```

```
    // Logic:
    // - [step-by-step breakdown of the condition]
    // - [how variables are used]
    // - [any transformations applied]
    // Success Path: [what happens when condition is true]
    // Failure Path: [what happens when condition is false]
    // Business Rules: [specific rules being enforced]
    // Error Handling: [how errors are managed]
    // Financial Impact: [effect on aid calculations]
    // If N/A, explicitly state: "CONDITIONALS: N/A"
  2. BOUNDARY CHECKS Section:
    Format:
    // BOUNDARY CHECKS:
    // Check [number]: [specific limit being enforced]
    // Source: [where boundary values come from]
    // Variables: [variables and methods involved]
    // Validation:
    // - [how boundaries are checked]
    // - [validation method details]
    // Edge Cases:
    // - [specific edge cases handled]
    // - [how each is managed]
    // Error Handling: [boundary violation management]
    // Impact: [effect of boundary violations]
    // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
  3. ARITHMETIC Section:
    Format:
    // ARITHMETIC:
    // Operation [number]: [name of calculation]
    // Formula: [complete mathematical expression]
    // Variables:
    // - [each variable and its purpose]
    // - [data types and units]
    // Steps:
    // 1. [detailed step-by-step breakdown]
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
  ADDITIONAL REQUIREMENTS:
  1. Complex Logic Sections:
    - Must break down multi-step operations
    - Explain interactions between components
    - Detail data flow and transformations
  2. Error Handling:
    - Document all error conditions
    - Explain recovery procedures
    - Detail logging and reporting
  3. API Interactions:
    - Document all platform method calls
    - Explain data transformations
    - Detail error handling
  4. Financial Aid Impact:
    - Explain effects on aid calculations
    - Document regulatory compliance
    - Detail student eligibility impact
  Example Section With All Aspects:
  // CONDITIONALS:
  // Check 1: Payment Period Status Validation
  // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
  // Purpose: Ensure only valid, active payment periods are included
  // Logic:
  // - Retrieve all loan payment periods
```

```
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.IsirVerificationClearanceRulesScript)
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
/*
Due to the changes to ISIR Verification for 24/25 we are no longer automatically triggering ISIR corrections
 within our
baseline configuration. An ISIR Verification task will be created once all acceptable documentation is
 received for a
Financial Aid Administrator to review. If Oracle determines that there is additional logic that can be added
 to increase
ISIR Verification Automation we will enhance our baseline configuration in future releases.
*/
//Note: if script returns true then auto code clear, if it returns false then create a task
//Student ISIR variables
def isirFPS_INCARCERATEDAPPLICANTFLAG =
 isirRecord.getField(IsirField.FPS_INCARCERATEDAPPLICANTFLAG).getOrNull()
def isirSTUDENT_FILED1040OR1040NR = isirRecord.getField(IsirField.STUDENT_FILED1040OR1040NR).getOrNull()
def isirSTUDENT_FILEDNONUSTAXRETURN = isirRecord.getField(IsirField.STUDENT_FILEDNONUSTAXRETURN).getOrNull()
def isirSTUDENTFTIM_IRSRESPONSECODE_FTI =
 isirRecord.getField(IsirField.STUDENTFTIM_IRSRESPONSECODE_FTI).getOrNull()
def isirSTUDENTFTIM_UNTAXEDIRADISTRIBUTIONS_FTI =
 isirRecord.getField(IsirField.STUDENTFTIM_UNTAXEDIRADISTRIBUTIONS_FTI).getOrNull()
def isirSTUDENTFTIM_IRADEDUCTIBLEANDPAYMENTS_FTI =
 isirRecord.getField(IsirField.STUDENTFTIM_IRADEDUCTIBLEANDPAYMENTS_FTI).getOrNull()
def isirSTUDENT_UNTAXEDPORTIONSOFIRADISTRIBUTIONS =
 isirRecord.getField(IsirField.STUDENT_UNTAXEDPORTIONSOFIRADISTRIBUTIONS).getOrNull()
def isirSTUDENT_IRAROLLOVER = isirRecord.getField(IsirField.STUDENT_IRAROLLOVER).getOrNull()
```
```

```
def isirSTUDENT_FOREIGNEARNEDINCOMEEXCLUSION =
 isirRecord.getField(IsirField.STUDENT_FOREIGNEARNEDINCOMEEXCLUSION).getOrNull()
//Parent ISIR variables
def isirPARENT_FILED1040OR1040NR = isirRecord.getField(IsirField.PARENT_FILED1040OR1040NR).getOrNull()
def isirPARENTFTIM_IRSRESPONSECODE_FTI =
 isirRecord.getField(IsirField.PARENTFTIM_IRSRESPONSECODE_FTI).getOrNull()
def isirPARENT_FILEDNONUSTAXRETURN = isirRecord.getField(IsirField.PARENT_FILEDNONUSTAXRETURN).getOrNull()
def isirPARENTFTIM_UNTAXEDIRADISTRIBUTIONS_FTI =
 isirRecord.getField(IsirField.PARENTFTIM_UNTAXEDIRADISTRIBUTIONS_FTI).getOrNull()
def isirPARENTFTIM_IRADEDUCTIBLEANDPAYMENTS_FTI =
 isirRecord.getField(IsirField.PARENTFTIM_IRADEDUCTIBLEANDPAYMENTS_FTI).getOrNull()
def isirPARENT_UNTAXEDPORTIONSOFIRADISTRIBUTIONS =
 isirRecord.getField(IsirField.PARENT_UNTAXEDPORTIONSOFIRADISTRIBUTIONS).getOrNull()
def isirPARENT_IRAROLLOVER = isirRecord.getField(IsirField.PARENT_IRAROLLOVER).getOrNull()
def isirPARENT_UPDATEDFAMILYSIZE = isirRecord.getField(IsirField.PARENT_UPDATEDFAMILYSIZE).getOrNull()
def isirFPS_ASSUMEDPARENTFAMILYSIZE = isirRecord.getField(IsirField.FPS_ASSUMEDPARENTFAMILYSIZE).getOrNull()
def isirPARENT_FOREIGNEARNEDINCOMEEXCLUSION =
 isirRecord.getField(IsirField.PARENT_FOREIGNEARNEDINCOMEEXCLUSION).getOrNull()
//Parent Spouse ISIR variables
def isirPARENTSPOUSE_IRAROLLOVER = isirRecord.getField(IsirField.PARENTSPOUSE_IRAROLLOVER).getOrNull()
def isirPARENTSPOUSE_UNTAXEDPORTIONSOFIRADISTRIBUTIONS =
 isirRecord.getField(IsirField.PARENTSPOUSE_UNTAXEDPORTIONSOFIRADISTRIBUTIONS).getOrNull()
def isirPARENTSPOUSEFTIM_IRADEDUCTIBLEANDPAYMENTS_FTI =
 isirRecord.getField(IsirField.PARENTSPOUSEFTIM_IRADEDUCTIBLEANDPAYMENTS_FTI).getOrNull()
def isirPARENTSPOUSEFTIM_UNTAXEDIRADISTRIBUTIONS_FTI =
 isirRecord.getField(IsirField.PARENTSPOUSEFTIM_UNTAXEDIRADISTRIBUTIONS_FTI).getOrNull()
def isirPARENTSPOUSE_FILED1040OR1040NR =
 isirRecord.getField(IsirField.PARENTSPOUSE_FILED1040OR1040NR).getOrNull()
def isirPARENTSPOUSEFTIM_IRSRESPONSECODE_FTI =
 isirRecord.getField(IsirField.PARENTSPOUSEFTIM_IRSRESPONSECODE_FTI).getOrNull()
def isirPARENTSPOUSE_FILEDNONUSTAXRETURN =
 isirRecord.getField(IsirField.PARENTSPOUSE_FILEDNONUSTAXRETURN).getOrNull()
def isirPARENTSPOUSE_TAXRETURNFILINGSTATUS =
 isirRecord.getField(IsirField.PARENTSPOUSE_TAXRETURNFILINGSTATUS).getOrNull()
def isirPARENTSPOUSE_FOREIGNEARNEDINCOMEEXCLUSION =
 isirRecord.getField(IsirField.PARENTSPOUSE_FOREIGNEARNEDINCOMEEXCLUSION).getOrNull()
//Custom documents received variables
boolean IdentificationDocsReceived = false
boolean SOEPDocsReceived = false
//Custom documents received check
if ((receivedDocuments.hasDoc("DriversLicense", "Student") && receivedDocuments.get("DriversLicense",
 "Student").isAcceptable()) || (receivedDocuments.hasDoc("NonDriversLicenseID",
 "Student") && receivedDocuments.get("NonDriversLicenseID", "Student").isAcceptable()) ||
 (receivedDocuments.hasDoc("StateIssuedID", "Student") && receivedDocuments.get("StateIssuedID",
 "Student").isAcceptable()) || (receivedDocuments.hasDoc("Passport", "Student") &&
 receivedDocuments.get("Passport", "Student").isAcceptable())) {
 IdentificationDocsReceived = true
}
if ((receivedDocuments.hasDoc("SOEP-Campus", "Student") && receivedDocuments.get("SOEP-
Campus", "Student").isAcceptable()) || (receivedDocuments.hasDoc("SOEP-Notary", "Student") &&
 receivedDocuments.get("SOEP-Notary", "Student").isAcceptable())) {
 SOEPDocsReceived = true
}
//If student is incarcerated and has custom documents then exit the script
if ((isirFPS_INCARCERATEDAPPLICANTFLAG == "1" || isirFPS_INCARCERATEDAPPLICANTFLAG == "2" ||
 isirFPS_INCARCERATEDAPPLICANTFLAG == "3") && IdentificationDocsReceived && SOEPDocsReceived) {
 return true
}
//Student auto code clearing logic
if ((isirSTUDENT_FILED1040OR1040NR == "1" || isirSTUDENT_FILED1040OR1040NR == null) && [null, "203", "206",
 "212", "214"].contains(isirSTUDENTFTIM_IRSRESPONSECODE_FTI)) {
 if ((receivedDocuments.hasDoc("1040", "Student") && receivedDocuments.get("1040",
 "Student").isAcceptable()) || (receivedDocuments.hasDoc("1040x", "Student") &&
 receivedDocuments.get("1040x", "Student").isAcceptable()) ||
 (receivedDocuments.hasDoc("TaxReturnTranscript", "Student") && receivedDocuments.get("TaxReturnTranscript",
 "Student").isAcceptable())) {
```

```
 return false
 }
}
if (isirSTUDENT_FILED1040OR1040NR == "2" && isirSTUDENT_FILEDNONUSTAXRETURN == "1") {
 if (receivedDocuments.hasDoc("ForeignTaxTranscript", "Student") &&
 receivedDocuments.get("ForeignTaxTranscript", "Student").isAcceptable()) {
 return false
 }
}
if (isirSTUDENT_FILED1040OR1040NR == "2" && (isirSTUDENT_FILEDNONUSTAXRETURN == "2" ||
 isirSTUDENT_FILEDNONUSTAXRETURN == null)) {
 if ((receivedDocuments.hasDoc("NonFilingStatement", "Student") &&
 receivedDocuments.get("NonFilingStatement", "Student").isAcceptable()) ||
 (receivedDocuments.hasDoc("W2", "Student") && receivedDocuments.get("W2", "Student").isAcceptable())
 || (receivedDocuments.hasDoc("1099G", "Student") && receivedDocuments.get("1099G",
 "Student").isAcceptable()) || (receivedDocuments.hasDoc("SelfEmploymentStatement", "Student") &&
 receivedDocuments.get("SelfEmploymentStatement", "Student").isAcceptable())) {
 return false
 }
}
//Parent auto code clearing logic
if ((isirPARENT_FILED1040OR1040NR == "1" || isirPARENT_FILED1040OR1040NR == null) && [null, "203", "206",
 "212", "214"].contains(isirPARENTFTIM_IRSRESPONSECODE_FTI)) {
 if ((receivedDocuments.hasDoc("1040", "Parent") && receivedDocuments.get("1040", "Parent").isAcceptable())
 || (receivedDocuments.hasDoc("1040x", "Parent") && receivedDocuments.get("1040x",
 "Parent").isAcceptable()) || (receivedDocuments.hasDoc("TaxReturnTranscript", "Parent") &&
 receivedDocuments.get("TaxReturnTranscript", "Parent").isAcceptable())) {
 return false
 }
}
if ((isirPARENT_FILED1040OR1040NR == "1" || isirPARENT_FILED1040OR1040NR == null) &&
 isirPARENT_FILEDNONUSTAXRETURN == "1") {
 if ((receivedDocuments.hasDoc("1040", "Parent") && receivedDocuments.get("1040", "Parent").isAcceptable())
 || (receivedDocuments.hasDoc("1040x", "Parent") && receivedDocuments.get("1040x",
 "Parent").isAcceptable()) || (receivedDocuments.hasDoc("TaxReturnTranscript", "Parent") &&
 receivedDocuments.get("TaxReturnTranscript", "Parent").isAcceptable())) {
 return false
 }
}
if (isirPARENT_FILED1040OR1040NR == "2" && isirPARENT_FILEDNONUSTAXRETURN == "2") {
 if (receivedDocuments.hasDoc("ForeignTaxTranscript", "Parent") &&
 receivedDocuments.get("ForeignTaxTranscript", "Parent").isAcceptable()) {
 return false
 }
}
if (isirPARENT_FILED1040OR1040NR == "2" && isirPARENT_FILEDNONUSTAXRETURN == "3") {
 if ((receivedDocuments.hasDoc("NonFilingStatement", "Parent") &&
 receivedDocuments.get("NonFilingStatement", "Parent").isAcceptable()) ||
 (receivedDocuments.hasDoc("ForeignTaxTranscript", "Parent") &&
 receivedDocuments.get("ForeignTaxTranscript", "Parent").isAcceptable())) {
 return false
 }
}
if (isirPARENT_FILED1040OR1040NR == "2" && ["4", "5", "6"].contains(isirPARENT_FILEDNONUSTAXRETURN)) {
 if ((receivedDocuments.hasDoc("NonFilingStatement", "Parent") &&
 receivedDocuments.get("NonFilingStatement", "Parent").isAcceptable()) ||
 (receivedDocuments.hasDoc("W2", "Parent") && receivedDocuments.get("W2", "Parent").isAcceptable())
 || (receivedDocuments.hasDoc("1099G", "Parent") && receivedDocuments.get("1099G",
 "Parent").isAcceptable()) || (receivedDocuments.hasDoc("SelfEmploymentStatement", "Parent") &&
 receivedDocuments.get("SelfEmploymentStatement", "Parent").isAcceptable())) {
 return false
 }
}
//Parent Spouse auto code clearing logic
```

```
if (isirPARENTSPOUSE_TAXRETURNFILINGSTATUS == "3" && (isirPARENTSPOUSE_FILED1040OR1040NR
 == "1" || isirPARENTSPOUSE_FILED1040OR1040NR == null) && [null, "203", "206", "212",
 "214"].contains(isirPARENTSPOUSEFTIM_IRSRESPONSECODE_FTI)) {
 if ((receivedDocuments.hasDoc("1040", "Parent Spouse") && receivedDocuments.get("1040",
 "Parent Spouse").isAcceptable()) || (receivedDocuments.hasDoc("1040x", "Parent
 Spouse") && receivedDocuments.get("1040x", "Parent Spouse").isAcceptable())
 || (receivedDocuments.hasDoc("TaxReturnTranscript", "Parent Spouse") &&
 receivedDocuments.get("TaxReturnTranscript", "Parent Spouse").isAcceptable())) {
 return false
 }
}
if (isirPARENTSPOUSE_TAXRETURNFILINGSTATUS == "3" && (isirPARENTSPOUSE_FILED1040OR1040NR == "1" ||
 isirPARENTSPOUSE_FILED1040OR1040NR == null) && isirPARENTSPOUSE_FILEDNONUSTAXRETURN == "1") {
 if ((receivedDocuments.hasDoc("1040", "Parent Spouse") && receivedDocuments.get("1040",
 "Parent Spouse").isAcceptable()) || (receivedDocuments.hasDoc("1040x", "Parent
 Spouse") && receivedDocuments.get("1040x", "Parent Spouse").isAcceptable())
 || (receivedDocuments.hasDoc("TaxReturnTranscript", "Parent Spouse") &&
 receivedDocuments.get("TaxReturnTranscript", "Parent Spouse").isAcceptable())) {
 return false
 }
}
if (isirPARENTSPOUSE_TAXRETURNFILINGSTATUS == "3" && isirPARENTSPOUSE_FILED1040OR1040NR == "2" &&
 isirPARENTSPOUSE_FILEDNONUSTAXRETURN == "2") {
 if (receivedDocuments.hasDoc("ForeignTaxTranscript", "Parent Spouse") &&
 receivedDocuments.get("ForeignTaxTranscript", "Parent Spouse").isAcceptable()) {
 return false
 }
}
if (isirPARENTSPOUSE_TAXRETURNFILINGSTATUS == "3" && isirPARENTSPOUSE_FILED1040OR1040NR == "2" &&
 isirPARENTSPOUSE_FILEDNONUSTAXRETURN == "3") {
 if ((receivedDocuments.hasDoc("NonFilingStatement", "Parent Spouse") &&
 receivedDocuments.get("NonFilingStatement", "Parent Spouse").isAcceptable())
 || (receivedDocuments.hasDoc("ForeignTaxTranscript", "Parent Spouse") &&
 receivedDocuments.get("ForeignTaxTranscript", "Parent Spouse").isAcceptable())) {
 return false
 }
}
if (isirPARENTSPOUSE_TAXRETURNFILINGSTATUS == "3" && isirPARENTSPOUSE_FILED1040OR1040NR == "2" && ["4", "5",
 "6"].contains(isirPARENTSPOUSE_FILEDNONUSTAXRETURN)) {
 if ((receivedDocuments.hasDoc("NonFilingStatement", "Parent Spouse") &&
 receivedDocuments.get("NonFilingStatement", "Parent Spouse").isAcceptable()) ||
 (receivedDocuments.hasDoc("W2", "Parent Spouse") && receivedDocuments.get("W2",
 "Parent Spouse").isAcceptable()) || (receivedDocuments.hasDoc("1099G", "Parent
 Spouse") && receivedDocuments.get("1099G", "Parent Spouse").isAcceptable())
 || (receivedDocuments.hasDoc("SelfEmploymentStatement", "Parent Spouse") &&
 receivedDocuments.get("SelfEmploymentStatement", "Parent Spouse").isAcceptable())) {
 return false
 }
}
//Student rollover information auto code clearing logic
if (isirSTUDENTFTIM_UNTAXEDIRADISTRIBUTIONS_FTI > 0 || isirSTUDENTFTIM_IRADEDUCTIBLEANDPAYMENTS_FTI > 0 ||
 isirSTUDENT_UNTAXEDPORTIONSOFIRADISTRIBUTIONS > 0 || isirSTUDENT_IRAROLLOVER > 0) {
 if (receivedDocuments.hasDoc("RolloverStatement", "Student") && receivedDocuments.get("RolloverStatement",
 "Student").isAcceptable()) {
 return false
 }
}
//Parent rollover information auto code clearing logic
if (isirPARENTFTIM_UNTAXEDIRADISTRIBUTIONS_FTI > 0 || isirPARENTFTIM_IRADEDUCTIBLEANDPAYMENTS_FTI > 0 ||
 isirPARENT_UNTAXEDPORTIONSOFIRADISTRIBUTIONS > 0 || isirPARENT_IRAROLLOVER > 0) {
 if (receivedDocuments.hasDoc("RolloverStatement", "Parent") && receivedDocuments.get("RolloverStatement",
 "Parent").isAcceptable()) {
 return false
 }
}
//Parent Spouse rollover information auto code clearing logic
```

```
if (isirPARENTSPOUSEFTIM_UNTAXEDIRADISTRIBUTIONS_FTI > 0 ||
 isirPARENTSPOUSEFTIM_IRADEDUCTIBLEANDPAYMENTS_FTI > 0 || isirPARENTSPOUSE_UNTAXEDPORTIONSOFIRADISTRIBUTIONS
 > 0 || isirPARENTSPOUSE_IRAROLLOVER > 0) {
 if (receivedDocuments.hasDoc("RolloverStatement", "Parent Spouse") &&
 receivedDocuments.get("RolloverStatement", "Parent Spouse").isAcceptable()) {
 return false
 }
}
//Student Foreign Income Exclusion auto code clearing logic
if (isirSTUDENT_FOREIGNEARNEDINCOMEEXCLUSION != null && (isirSTUDENT_FOREIGNEARNEDINCOMEEXCLUSION > 0 ||
 isirSTUDENT_FOREIGNEARNEDINCOMEEXCLUSION < 0)) {
 if ((receivedDocuments.hasDoc("1040", "Student") && receivedDocuments.get("1040",
 "Student").isAcceptable()) || (receivedDocuments.hasDoc("1040x", "Student") &&
 receivedDocuments.get("1040x", "Student").isAcceptable()) ||
 (receivedDocuments.hasDoc("TaxReturnTranscript", "Student") && receivedDocuments.get("TaxReturnTranscript",
 "Student").isAcceptable()) || (receivedDocuments.hasDoc("ForeignTaxTranscript", "Student") &&
 receivedDocuments.get("ForeignTaxTranscript", "Student").isAcceptable())) {
 return false
 }
}
//Parent Foreign Income Exclusion auto code clearing logic
if (isirPARENT_FOREIGNEARNEDINCOMEEXCLUSION != null && (isirPARENT_FOREIGNEARNEDINCOMEEXCLUSION > 0 ||
 isirPARENT_FOREIGNEARNEDINCOMEEXCLUSION < 0)) {
 if ((receivedDocuments.hasDoc("1040", "Parent") && receivedDocuments.get("1040", "Parent").isAcceptable())
 || (receivedDocuments.hasDoc("1040x", "Parent") && receivedDocuments.get("1040x",
 "Parent").isAcceptable()) || (receivedDocuments.hasDoc("TaxReturnTranscript",
 "Parent") && receivedDocuments.get("TaxReturnTranscript", "Parent").isAcceptable()) ||
 (receivedDocuments.hasDoc("ForeignTaxTranscript", "Parent") &&
 receivedDocuments.get("ForeignTaxTranscript", "Parent").isAcceptable())) {
 return false
 }
}
//Parent Spouse Foreign Income Exclusion auto code clearing logic
if (isirPARENTSPOUSE_FOREIGNEARNEDINCOMEEXCLUSION != null && (isirPARENTSPOUSE_FOREIGNEARNEDINCOMEEXCLUSION
 > 0 || isirPARENTSPOUSE_FOREIGNEARNEDINCOMEEXCLUSION < 0)) {
 if ((receivedDocuments.hasDoc("1040", "Parent Spouse") && receivedDocuments.get("1040",
 "Parent Spouse").isAcceptable()) || (receivedDocuments.hasDoc("1040x", "Parent
 Spouse") && receivedDocuments.get("1040x", "Parent Spouse").isAcceptable())
 || (receivedDocuments.hasDoc("TaxReturnTranscript", "Parent Spouse") &&
 receivedDocuments.get("TaxReturnTranscript", "Parent Spouse").isAcceptable())
 || (receivedDocuments.hasDoc("ForeignTaxTranscript", "Parent Spouse") &&
 receivedDocuments.get("ForeignTaxTranscript", "Parent Spouse").isAcceptable())) {
 return false
 }
}
//Family size auto code clearing logic
if (isirPARENT_UPDATEDFAMILYSIZE != null || isirFPS_ASSUMEDPARENTFAMILYSIZE != null) {
 if (receivedDocuments.hasDoc("VW-Dep", "Student") && receivedDocuments.get("VW-Dep",
 "Student").isAcceptable()) {
 return false
 }
}
return true
```
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
```

```
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# ISIR Use Subsequent ISIR

Name: isirUseSubsequentIsir

Purpose: Define which fields on ISIR will trigger an ISIR review when subsequent ISIRs enter SFP for a Student.

In Baseline Config: Yes

Return Type: Boolean

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "isirUseSubsequentIsir"
Primary Function: Define which fields on ISIR will trigger an ISIR review when subsequent ISIRs enter SFP
 for a Student.
Expected Output Type: Boolean
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
```

```
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
latestIsirRecord
Purpose: The student's latest ISIR record.
Type: IIsirRecordAPI
Available Methods:
* getActiveIsirForAwardYear
 - Returns: Boolean
 - Function: Gets the active ISIR for award year status.'
* getAllFields
 - Returns: @NonNull Map<IsirField<?>,IIsirFieldValue<?>>
 - Function: Retrieves all field values (WARNING: can only be used with 2025+ ISIRs).'
* getAssumptions
 - Returns: List<IAssumptionAPI>
 - Function: Get the assumptions.'
* getAwardYear
 - Returns: String
 - Function: Gets the award year.'
* getCommentCodes
 - Returns: List<ICommentCodeAPI>
 - Function: Get the comment codes.'
* getCpsTransNumber
 - Returns: String
 - Function: Gets the CPS transaction number.'
* getCurrentSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's current SSN.'
* getDiscrepancies
 - Returns: List<IDiscrepancyAPI>
 - Function: Get the discrepancies.'
* getField
 - Returns: <T> @NonNull IIsirFieldValue<T>
 - Function: Retrieves the value of a given field (WARNING: can only be used with 2025+ ISIRs).'
* getFirstName
 - Returns: String
 - Function: Gets the student's first name.'
* getHighlightFlags
 - Returns: List<IHighlightFlagAPI>
 - Function: Get the highlight flags.'
* getIsirFieldValue
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValueOrNull
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValues
 - Returns: Map<String,String>
 - Function: Get a map containing all ISIR fields (WARNING: This method can only be used up to 2024 award
 year).'
* getIsirIdOriginalNameId
 - Returns: String
 - Function: Gets the student's original name ID.'
```

```
* getIsirIdSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's original SSN.'
* getIsirIdYearIndicator
 - Returns: String
 - Function: Gets the ISIR's year indicator.'
* getIsirStatus
 - Returns: String
 - Function: Gets the ISIR status.'
* getIsirStatusBeforeCorrection
 - Returns: String
 - Function: When ISIR Status is Corrected returns the previous ISIR Status.'
* getLastName
 - Returns: String
 - Function: Gets the student's last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the student's middle name.'
* getReceivedDate
 - Returns: Date
 - Function: Gets the system receipt date of the ISIR.'
* getRejectCodes
 - Returns: List<IRejectCodeAPI>
 - Function: Get the reject codes.'
* getSchoolCode1
 - Returns: String
 - Function: Gets the school code 1.'
* getSchoolCode10
 - Returns: String
 - Function: Gets the school code 10.'
* getSchoolCode2
 - Returns: String
 - Function: Gets the school code 2.'
* getSchoolCode3
 - Returns: String
 - Function: Gets the school code 3.'
* getSchoolCode4
 - Returns: String
 - Function: Gets the school code 4.'
* getSchoolCode5
 - Returns: String
 - Function: Gets the school code 5.'
* getSchoolCode6
 - Returns: String
 - Function: Gets the school code 6.'
* getSchoolCode7
 - Returns: String
 - Function: Gets the school code 7.'
* getSchoolCode8
 - Returns: String
 - Function: Gets the school code 8.'
* getSchoolCode9
 - Returns: String
 - Function: Gets the school code 9.'
* isCompatibleWithPackaging
 - Returns: boolean
 - Function: Checks if this ISIR satisfies minimal acceptance criteria for automatic Financial
Aid Package Calculation.'
* isUnverified
 - Returns: boolean
 - Function: Gets the ISIR verification status.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
```

```
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
previousIsirRecord
Purpose: The student's previous ISIR record.
Type: IIsirRecordAPI
Available Methods:
* getActiveIsirForAwardYear
 - Returns: Boolean
 - Function: Gets the active ISIR for award year status.'
* getAllFields
 - Returns: @NonNull Map<IsirField<?>,IIsirFieldValue<?>>
 - Function: Retrieves all field values (WARNING: can only be used with 2025+ ISIRs).'
* getAssumptions
 - Returns: List<IAssumptionAPI>
 - Function: Get the assumptions.'
* getAwardYear
 - Returns: String
 - Function: Gets the award year.'
* getCommentCodes
 - Returns: List<ICommentCodeAPI>
 - Function: Get the comment codes.'
* getCpsTransNumber
 - Returns: String
 - Function: Gets the CPS transaction number.'
* getCurrentSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's current SSN.'
* getDiscrepancies
 - Returns: List<IDiscrepancyAPI>
 - Function: Get the discrepancies.'
* getField
 - Returns: <T> @NonNull IIsirFieldValue<T>
 - Function: Retrieves the value of a given field (WARNING: can only be used with 2025+ ISIRs).'
* getFirstName
 - Returns: String
```

```
 - Function: Gets the student's first name.'
* getHighlightFlags
 - Returns: List<IHighlightFlagAPI>
 - Function: Get the highlight flags.'
* getIsirFieldValue
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValueOrNull
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValues
 - Returns: Map<String,String>
 - Function: Get a map containing all ISIR fields (WARNING: This method can only be used up to 2024 award
 year).'
* getIsirIdOriginalNameId
 - Returns: String
 - Function: Gets the student's original name ID.'
* getIsirIdSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's original SSN.'
* getIsirIdYearIndicator
 - Returns: String
 - Function: Gets the ISIR's year indicator.'
* getIsirStatus
 - Returns: String
 - Function: Gets the ISIR status.'
* getIsirStatusBeforeCorrection
 - Returns: String
 - Function: When ISIR Status is Corrected returns the previous ISIR Status.'
* getLastName
 - Returns: String
 - Function: Gets the student's last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the student's middle name.'
* getReceivedDate
 - Returns: Date
 - Function: Gets the system receipt date of the ISIR.'
* getRejectCodes
 - Returns: List<IRejectCodeAPI>
 - Function: Get the reject codes.'
* getSchoolCode1
 - Returns: String
 - Function: Gets the school code 1.'
* getSchoolCode10
 - Returns: String
 - Function: Gets the school code 10.'
* getSchoolCode2
 - Returns: String
 - Function: Gets the school code 2.'
* getSchoolCode3
 - Returns: String
 - Function: Gets the school code 3.'
* getSchoolCode4
 - Returns: String
 - Function: Gets the school code 4.'
* getSchoolCode5
 - Returns: String
 - Function: Gets the school code 5.'
* getSchoolCode6
 - Returns: String
 - Function: Gets the school code 6.'
* getSchoolCode7
 - Returns: String
 - Function: Gets the school code 7.'
* getSchoolCode8
```

```
 - Returns: String
 - Function: Gets the school code 8.'
* getSchoolCode9
 - Returns: String
 - Function: Gets the school code 9.'
* isCompatibleWithPackaging
 - Returns: boolean
 - Function: Checks if this ISIR satisfies minimal acceptance criteria for automatic Financial
Aid Package Calculation.'
* isUnverified
 - Returns: boolean
 - Function: Gets the ISIR verification status.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
```

```
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
```

```
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
```

```
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
```

```
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
awardYearPellTables
Purpose: PELL tables.
Type: IAwardYearPellTablesAPI
Available Methods:
* forAwardYear
 - Returns: IPellTableAPI
 - Function: Returns the collection of all Pell Tables (IPellTableAPI) for an award year on an on-demand,
cached basis.'
awy
Purpose: The award year of the student's latest ISIR record.
Type: Integer
Available Methods:
No methods available
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
```

```
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
```

```
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
```

```
matter the priority.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
```

```
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
```

```
    // Success Path: [what happens when condition is true]
    // Failure Path: [what happens when condition is false]
    // Business Rules: [specific rules being enforced]
    // Error Handling: [how errors are managed]
    // Financial Impact: [effect on aid calculations]
    // If N/A, explicitly state: "CONDITIONALS: N/A"
   2. BOUNDARY CHECKS Section:
    Format:
    // BOUNDARY CHECKS:
    // Check [number]: [specific limit being enforced]
    // Source: [where boundary values come from]
    // Variables: [variables and methods involved]
    // Validation:
    // - [how boundaries are checked]
    // - [validation method details]
    // Edge Cases:
    // - [specific edge cases handled]
    // - [how each is managed]
    // Error Handling: [boundary violation management]
    // Impact: [effect of boundary violations]
    // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
   3. ARITHMETIC Section:
    Format:
    // ARITHMETIC:
    // Operation [number]: [name of calculation]
    // Formula: [complete mathematical expression]
    // Variables:
    // - [each variable and its purpose]
    // - [data types and units]
    // Steps:
    // 1. [detailed step-by-step breakdown]
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
   ADDITIONAL REQUIREMENTS:
   1. Complex Logic Sections:
    - Must break down multi-step operations
    - Explain interactions between components
    - Detail data flow and transformations
   2. Error Handling:
    - Document all error conditions
    - Explain recovery procedures
    - Detail logging and reporting
   3. API Interactions:
    - Document all platform method calls
    - Explain data transformations
    - Detail error handling
   4. Financial Aid Impact:
    - Explain effects on aid calculations
    - Document regulatory compliance
    - Detail student eligibility impact
   Example Section With All Aspects:
   // CONDITIONALS:
   // Check 1: Payment Period Status Validation
   // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
   // Purpose: Ensure only valid, active payment periods are included
   // Logic:
   // - Retrieve all loan payment periods
   // - Filter based on status (!= "CANCELED")
   // - Compare start dates with reference date
   // Success Path: Period included in calculations
   // Failure Path: Period excluded from totals
```

```
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.IsirUseSubsequentIsirScript)
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
def isirTXID_TRANSACTIONSOURCE = latestIsirRecord.getField(IsirField.TXID_TRANSACTIONSOURCE).getOrNull()
if (isirTXID_TRANSACTIONSOURCE != null) {
 return true
}
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# ISIR Discrepancy Evaluation

Name: isirDiscrepancyEvaluation

Purpose: Determine whether an ISIR correction for a given ISIR field is required.

In Baseline Config: Yes

Return Type: Void

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "isirDiscrepancyEvaluation"
Primary Function: Determine whether an ISIR correction for a given ISIR field is required, and if so, what
 value from what source to
correct the ISIR field to.
Expected Output Type: Void
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
allAwardYearReceivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
```

```
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
```

```
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
discrepancyEvaluationResult
Purpose: Binding that shall be populated by the script with the discrepancy evaluation result.
Type: IDiscrepancyEvaluationResultAPI
Available Methods:
* getStudentFollowUpReason
 - Returns: String
 - Function: Gets the student follow-up reason.'
* isNeedManualReview
 - Returns: boolean
 - Function: Gets the manual review status.'
* getAssumptionOverride
 - Returns: String
 - Function: Gets the assumption override.'
* getCorrectedValue
 - Returns: String
 - Function: Gets the corrected value.'
* getEmailAddress
 - Returns: String
 - Function: Gets the email address.'
* getHasDiscrepancy
 - Returns: boolean
 - Function: Gets the discrepancy existence status.'
* getHighSchoolName
```

```
    - Returns: String
    - Function: Gets the high school name.'
  * getMessage
    - Returns: String
    - Function: Gets the discrepancy message.'
  * getParentsEmailAddress
    - Returns: String
    - Function: Gets the parent's email address.'
  * getRejectOverride
    - Returns: String
    - Function: Gets the reject override.'
  * isStudentFollowUpRequired
    - Returns: boolean
    - Function: Gets the student follow-up required status.'
  * setAssumptionOverride
    - Returns: void
    - Function: Sets the assumption override.'
  * setCorrectedValue
    - Returns: void
    - Function: Sets the corrected value.'
  * setEmailAddress
    - Returns: void
    - Function: Sets the email address.'
  * setHasDiscrepancy
    - Returns: void
    - Function: Sets the discrepancy existence status.'
  * setHighSchoolName
    - Returns: void
    - Function: Sets the high school name.'
  * setMessage
    - Returns: void
    - Function: Sets the discrepancy message.'
  * setNeedManualReview
    - Returns: void
    - Function: Sets the manual review status.'
  * setParentsEmailAddress
    - Returns: void
    - Function: Sets the parent's email address.'
  * setRejectOverride
    - Returns: void
    - Function: Sets the reject override.'
  * setStudentFollowUpReason
    - Returns: void
    - Function: Sets the student follow-up reason.'
  * setStudentFollowUpRequired
    - Returns: void
    - Function: Sets the student follow-up required status.'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
  * getLifetimeLimit
    - Returns: int
    - Function: No description available'
  * getRemainingNsldsLimit
    - Returns: int
    - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
  * belongsTo
    - Returns: boolean
    - Function: Checks if this enrollment belongs to a student.'
  * getBindings
    - Returns: IEnrollmentBindingsAPI
```

```
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
```

```
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
```

```
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isirRecord
Purpose: The ISIR being evaluated.
Type: IIsirRecordAPI
Available Methods:
* getActiveIsirForAwardYear
 - Returns: Boolean
 - Function: Gets the active ISIR for award year status.'
* getAllFields
 - Returns: @NonNull Map<IsirField<?>,IIsirFieldValue<?>>
 - Function: Retrieves all field values (WARNING: can only be used with 2025+ ISIRs).'
* getAssumptions
 - Returns: List<IAssumptionAPI>
 - Function: Get the assumptions.'
* getAwardYear
 - Returns: String
 - Function: Gets the award year.'
* getCommentCodes
 - Returns: List<ICommentCodeAPI>
 - Function: Get the comment codes.'
* getCpsTransNumber
 - Returns: String
 - Function: Gets the CPS transaction number.'
* getCurrentSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's current SSN.'
* getDiscrepancies
 - Returns: List<IDiscrepancyAPI>
 - Function: Get the discrepancies.'
* getField
 - Returns: <T> @NonNull IIsirFieldValue<T>
 - Function: Retrieves the value of a given field (WARNING: can only be used with 2025+ ISIRs).'
* getFirstName
 - Returns: String
 - Function: Gets the student's first name.'
* getHighlightFlags
 - Returns: List<IHighlightFlagAPI>
 - Function: Get the highlight flags.'
* getIsirFieldValue
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValueOrNull
```

```
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValues
 - Returns: Map<String,String>
 - Function: Get a map containing all ISIR fields (WARNING: This method can only be used up to 2024 award
 year).'
* getIsirIdOriginalNameId
 - Returns: String
 - Function: Gets the student's original name ID.'
* getIsirIdSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's original SSN.'
* getIsirIdYearIndicator
 - Returns: String
 - Function: Gets the ISIR's year indicator.'
* getIsirStatus
 - Returns: String
 - Function: Gets the ISIR status.'
* getIsirStatusBeforeCorrection
 - Returns: String
 - Function: When ISIR Status is Corrected returns the previous ISIR Status.'
* getLastName
 - Returns: String
 - Function: Gets the student's last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the student's middle name.'
* getReceivedDate
 - Returns: Date
 - Function: Gets the system receipt date of the ISIR.'
* getRejectCodes
 - Returns: List<IRejectCodeAPI>
 - Function: Get the reject codes.'
* getSchoolCode1
 - Returns: String
 - Function: Gets the school code 1.'
* getSchoolCode10
 - Returns: String
 - Function: Gets the school code 10.'
* getSchoolCode2
 - Returns: String
 - Function: Gets the school code 2.'
* getSchoolCode3
 - Returns: String
 - Function: Gets the school code 3.'
* getSchoolCode4
 - Returns: String
 - Function: Gets the school code 4.'
* getSchoolCode5
 - Returns: String
 - Function: Gets the school code 5.'
* getSchoolCode6
 - Returns: String
 - Function: Gets the school code 6.'
* getSchoolCode7
 - Returns: String
 - Function: Gets the school code 7.'
* getSchoolCode8
 - Returns: String
 - Function: Gets the school code 8.'
* getSchoolCode9
 - Returns: String
 - Function: Gets the school code 9.'
* isCompatibleWithPackaging
 - Returns: boolean
 - Function: Checks if this ISIR satisfies minimal acceptance criteria for automatic Financial
```

```
Aid Package Calculation.'
* isUnverified
 - Returns: boolean
 - Function: Gets the ISIR verification status.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
previousIsirRecords
Purpose: Previous ISIR records with the same award year as the ISIR being evaluated, ordered by descending
 CPS numbers.
Type: List
Available Methods:
No methods available
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
```

```
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Active documents for the student for the award year of the ISIR being evaluated.
Type: IReceivedDocs
Available Methods:
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and Student, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and owners, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code, owners, and subtype, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and Student, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and owners, if it exists.'
* getValidDocCodes
 - Returns: Set<String>
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: Return true if doc exists with the right code and owners'
* noDocs
 - Returns: boolean
 - Function: No description available'
* put
 - Returns: void
```

```
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
```

```
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
throwsDocumentIsFaUnacceptableExceptionParam
Purpose: This binding can be set within a Groovy script to control whether a subsequent call to
IReceivedDoc.isAcceptable() or IReceivedDocDocumentAPI.isAcceptable() triggers an exception or not when a
document is unnaceptable. The default value is true.
Type: Boolean
Available Methods:
No methods available
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
```

```
    // Source: [where boundary values come from]
    // Variables: [variables and methods involved]
    // Validation:
    // - [how boundaries are checked]
    // - [validation method details]
    // Edge Cases:
    // - [specific edge cases handled]
    // - [how each is managed]
    // Error Handling: [boundary violation management]
    // Impact: [effect of boundary violations]
    // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
  3. ARITHMETIC Section:
    Format:
    // ARITHMETIC:
    // Operation [number]: [name of calculation]
    // Formula: [complete mathematical expression]
    // Variables:
    // - [each variable and its purpose]
    // - [data types and units]
    // Steps:
    // 1. [detailed step-by-step breakdown]
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
  ADDITIONAL REQUIREMENTS:
  1. Complex Logic Sections:
    - Must break down multi-step operations
    - Explain interactions between components
    - Detail data flow and transformations
  2. Error Handling:
    - Document all error conditions
    - Explain recovery procedures
    - Detail logging and reporting
  3. API Interactions:
    - Document all platform method calls
    - Explain data transformations
    - Detail error handling
  4. Financial Aid Impact:
    - Explain effects on aid calculations
    - Document regulatory compliance
    - Detail student eligibility impact
  Example Section With All Aspects:
  // CONDITIONALS:
  // Check 1: Payment Period Status Validation
  // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
  // Purpose: Ensure only valid, active payment periods are included
  // Logic:
  // - Retrieve all loan payment periods
  // - Filter based on status (!= "CANCELED")
  // - Compare start dates with reference date
  // Success Path: Period included in calculations
  // Failure Path: Period excluded from totals
  // Business Rules: Canceled periods must not affect grade level
  // Error Handling: Invalid periods logged and skipped
  // Financial Impact: Affects available loan amounts
  //
  // BOUNDARY CHECKS:
  // Check 1: Payment Period Date Validation
  // Source: Academic Calendar Configuration
  // Variables: period.getStartDate(), referenceDate
  // Validation:
  // - Compare period start date with reference date
```

```
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.IsirDiscrepancyEvaluationScript)
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
//Variable definitions
def isirSPOUSE_FILED1040OR1040NR = isirRecord.getField(IsirField.SPOUSE_FILED1040OR1040NR).getOrNull()
def docValue = ""
//1040 discrepancy rules
if (receivedDocuments.hasDoc("1040","Student Spouse") && receivedDocuments.get("1040","Student
 Spouse").isAcceptable()) {
 if (receivedDocuments.get("1040", "Student Spouse").getDocumentField("AC1084") != null && !
receivedDocuments.get("1040", "Student Spouse").getDocumentField("AC1084").isAllWhitespace()) {
 docValue = receivedDocuments.get("1040", "Student Spouse").getDocumentField("AC1084")
 if ((docValue == "1040" || docValue == "1040NR") && (isirSPOUSE_FILED1040OR1040NR == "2" ||
 isirSPOUSE_FILED1040OR1040NR == null)) {
 discrepancyEvaluationResult.hasDiscrepancy = true
 discrepancyEvaluationResult.correctedValue = "1"
 return
 }
 }
}
//1040x discrepancy rules
if (receivedDocuments.hasDoc("1040x","Student Spouse") && receivedDocuments.get("1040x","Student
 Spouse").isAcceptable()) {
 if (receivedDocuments.get("1040x", "Student Spouse").getDocumentField("AC1084") != null && !
receivedDocuments.get("1040x", "Student Spouse").getDocumentField("AC1084").isAllWhitespace()) {
 docValue = receivedDocuments.get("1040x", "Student Spouse").getDocumentField("AC1084")
 if ((docValue == "1040" || docValue == "1040NR") && (isirSPOUSE_FILED1040OR1040NR == "2" ||
 isirSPOUSE_FILED1040OR1040NR == null)) {
 discrepancyEvaluationResult.hasDiscrepancy = true
 discrepancyEvaluationResult.correctedValue = "1"
 return
 }
 }
}
//Tax Transcript discrepancy rules
if (receivedDocuments.hasDoc("TaxReturnTranscript","Student Spouse") &&
 receivedDocuments.get("TaxReturnTranscript","Student Spouse").isAcceptable()) {
 if (receivedDocuments.get("TaxReturnTranscript", "Student Spouse").getDocumentField("AC1084") != null && !
receivedDocuments.get("TaxReturnTranscript", "Student Spouse").getDocumentField("AC1084").isAllWhitespace())
 {
 docValue = receivedDocuments.get("TaxReturnTranscript", "Student Spouse").getDocumentField("AC1084")
```
```

```
 if ((docValue == "1040" || docValue == "1040NR") && (isirSPOUSE_FILED1040OR1040NR == "2" ||
 isirSPOUSE_FILED1040OR1040NR == null)) {
 discrepancyEvaluationResult.hasDiscrepancy = true
 discrepancyEvaluationResult.correctedValue = "1"
 return
 }
 }
}
```
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# ISIR Highlights Documents Required

Name: isirHighlightsDocumentsRequired

Purpose: Determine the documents required to resolve an ISIR highlight.

In Baseline Config: No

Return Type: Void

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "isirHighlightsDocumentsRequired"
Primary Function: Determine documents required to resolve an ISIR highlight.
Expected Output Type: Void
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
AGGREGATES_PERKINS_FLAGS
```

```
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
GRANT_PAYMENT
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
LOAN_DETAIL
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
OVERPAYMENT
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
allAwardYearReceivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
```

```
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
```

```
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
documentRequest
Purpose: Binding that shall be populated by the script with the selected documents.
Type: IIsirRequiredDocsAPI
Available Methods:
* addAnyDocuments
 - Returns: void
 - Function: Adds a list of documents, any one in the list is required.'
* addAnyDocuments
 - Returns: void
 - Function: Adds a list of documents, any one in the list is required.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* any
 - Returns: IIsirRequiredDocsAPI
 - Function: No description available'
* clear
 - Returns: void
 - Function: Clears all docs.'
* doc
 - Returns: IIsirRequiredDocsAPI
 - Function: No description available'
* getAdditionalDocRequestMessages
 - Returns: List<String>
 - Function: No description available'
* getValidDocCodes
 - Returns: Set<String>
 - Function: No description available'
* hasDocument
```

```
 - Returns: boolean
 - Function: Returns true if the document has been added, either as a regular request
or an 'any' request.'
* hasDocument
 - Returns: boolean
 - Function: Returns true if the document has been added, either as a regular request
or an 'any' request.'
* isEmpty
 - Returns: boolean
 - Function: Checks if these requirements are empty.'
* setValidDocCodes
 - Returns: void
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
```

```
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
```

```
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isirRecord
Purpose: The ISIR being evaluated.
Type: IIsirRecordAPI
Available Methods:
* getActiveIsirForAwardYear
 - Returns: Boolean
 - Function: Gets the active ISIR for award year status.'
* getAllFields
 - Returns: @NonNull Map<IsirField<?>,IIsirFieldValue<?>>
 - Function: Retrieves all field values (WARNING: can only be used with 2025+ ISIRs).'
* getAssumptions
 - Returns: List<IAssumptionAPI>
 - Function: Get the assumptions.'
* getAwardYear
 - Returns: String
 - Function: Gets the award year.'
* getCommentCodes
 - Returns: List<ICommentCodeAPI>
 - Function: Get the comment codes.'
```

```
* getCpsTransNumber
 - Returns: String
 - Function: Gets the CPS transaction number.'
* getCurrentSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's current SSN.'
* getDiscrepancies
 - Returns: List<IDiscrepancyAPI>
 - Function: Get the discrepancies.'
* getField
 - Returns: <T> @NonNull IIsirFieldValue<T>
 - Function: Retrieves the value of a given field (WARNING: can only be used with 2025+ ISIRs).'
* getFirstName
 - Returns: String
 - Function: Gets the student's first name.'
* getHighlightFlags
 - Returns: List<IHighlightFlagAPI>
 - Function: Get the highlight flags.'
* getIsirFieldValue
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValueOrNull
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValues
 - Returns: Map<String,String>
 - Function: Get a map containing all ISIR fields (WARNING: This method can only be used up to 2024 award
 year).'
* getIsirIdOriginalNameId
 - Returns: String
 - Function: Gets the student's original name ID.'
* getIsirIdSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's original SSN.'
* getIsirIdYearIndicator
 - Returns: String
 - Function: Gets the ISIR's year indicator.'
* getIsirStatus
 - Returns: String
 - Function: Gets the ISIR status.'
* getIsirStatusBeforeCorrection
 - Returns: String
 - Function: When ISIR Status is Corrected returns the previous ISIR Status.'
* getLastName
 - Returns: String
 - Function: Gets the student's last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the student's middle name.'
* getReceivedDate
 - Returns: Date
 - Function: Gets the system receipt date of the ISIR.'
* getRejectCodes
 - Returns: List<IRejectCodeAPI>
 - Function: Get the reject codes.'
* getSchoolCode1
 - Returns: String
 - Function: Gets the school code 1.'
* getSchoolCode10
 - Returns: String
 - Function: Gets the school code 10.'
* getSchoolCode2
 - Returns: String
 - Function: Gets the school code 2.'
* getSchoolCode3
 - Returns: String
```

```
 - Function: Gets the school code 3.'
* getSchoolCode4
 - Returns: String
 - Function: Gets the school code 4.'
* getSchoolCode5
 - Returns: String
 - Function: Gets the school code 5.'
* getSchoolCode6
 - Returns: String
 - Function: Gets the school code 6.'
* getSchoolCode7
 - Returns: String
 - Function: Gets the school code 7.'
* getSchoolCode8
 - Returns: String
 - Function: Gets the school code 8.'
* getSchoolCode9
 - Returns: String
 - Function: Gets the school code 9.'
* isCompatibleWithPackaging
 - Returns: boolean
 - Function: Checks if this ISIR satisfies minimal acceptance criteria for automatic Financial
Aid Package Calculation.'
* isUnverified
 - Returns: boolean
 - Function: Gets the ISIR verification status.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
nsldsGuarantorServicers
Purpose: List of all guarantors and servicers.
Type: IGuarantorServicersListAPI
Available Methods:
* getAwardYear
 - Returns: String
 - Function: No description available'
* getGuarantorServicerName
 - Returns: String
 - Function: No description available'
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
```

```
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
```

```
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
```

```
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Active documents for the student for the award year of the ISIR being evaluated.
Type: IReceivedDocs
Available Methods:
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and Student, if it exists.'
* get
 - Returns: IReceivedDoc
```

```
 - Function: Return doc for specified code and owners, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code, owners, and subtype, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and Student, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and owners, if it exists.'
* getValidDocCodes
 - Returns: Set<String>
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: Return true if doc exists with the right code and owners'
* noDocs
 - Returns: boolean
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
recentFahRecords
Purpose: All Financial Aid History records associated with the student.
Type: IRecentFahRecordsAPI
Available Methods:
* hasRecordsOfType
 - Returns: boolean
 - Function: No description available'
* overlapsLoan
 - Returns: boolean
 - Function: No description available'
* getRecords
 - Returns: List<IFahRecordAPI>
 - Function: No description available'
* getRecordsByType
 - Returns: List<IFahRecordAPI>
 - Function: No description available'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
```

```
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
throwsDocumentIsFaUnacceptableExceptionParam
Purpose: This binding can be set within a Groovy script to control whether a subsequent call to
IReceivedDoc.isAcceptable() or IReceivedDocDocumentAPI.isAcceptable() triggers an exception or not when a
document is unnaceptable. The default value is true.
Type: Boolean
Available Methods:
No methods available
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
```

```
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
 // Operation [number]: [name of calculation]
 // Formula: [complete mathematical expression]
 // Variables:
 // - [each variable and its purpose]
 // - [data types and units]
 // Steps:
 // 1. [detailed step-by-step breakdown]
 // 2. [intermediate calculations]
 // 3. [transformations applied]
 // Business Context: [why this calculation is needed]
 // Result: [meaning of the output]
 // Financial Impact: [effect on aid calculations]
```

```
 // Limitations: [any mathematical constraints]
 // If N/A, explicitly state: "ARITHMETIC: N/A"
ADDITIONAL REQUIREMENTS:
1. Complex Logic Sections:
 - Must break down multi-step operations
 - Explain interactions between components
 - Detail data flow and transformations
2. Error Handling:
 - Document all error conditions
 - Explain recovery procedures
 - Detail logging and reporting
3. API Interactions:
 - Document all platform method calls
 - Explain data transformations
 - Detail error handling
4. Financial Aid Impact:
 - Explain effects on aid calculations
 - Document regulatory compliance
 - Detail student eligibility impact
Example Section With All Aspects:
// CONDITIONALS:
// Check 1: Payment Period Status Validation
// Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
N/A - There is no baseline config. This is a placeholder.
```
## COMMENT GENERATION REQUIREMENTS:
```

```
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# ISIR C-Code Documents Required

Name: isirCCodeDocumentsRequired

Purpose: Determine the documents required to resolve an ISIR c-code.

In Baseline Config: No

Return Type: Void

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "isirCCodeDocumentsRequired"
Primary Function: Determine documents required to resolve an ISIR c-code.
Expected Output Type: Void
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
AGGREGATES_PERKINS_FLAGS
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
GRANT_PAYMENT
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
LOAN_DETAIL
```

```
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
OVERPAYMENT
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
allAwardYearReceivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
```

```
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
```

```
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
documentRequest
Purpose: Binding that shall be populated by the script with the selected documents.
Type: IIsirRequiredDocsAPI
Available Methods:
* addAnyDocuments
 - Returns: void
 - Function: Adds a list of documents, any one in the list is required.'
* addAnyDocuments
 - Returns: void
 - Function: Adds a list of documents, any one in the list is required.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* any
 - Returns: IIsirRequiredDocsAPI
 - Function: No description available'
* clear
 - Returns: void
 - Function: Clears all docs.'
* doc
 - Returns: IIsirRequiredDocsAPI
 - Function: No description available'
* getAdditionalDocRequestMessages
 - Returns: List<String>
 - Function: No description available'
* getValidDocCodes
 - Returns: Set<String>
 - Function: No description available'
* hasDocument
 - Returns: boolean
 - Function: Returns true if the document has been added, either as a regular request
or an 'any' request.'
* hasDocument
 - Returns: boolean
 - Function: Returns true if the document has been added, either as a regular request
or an 'any' request.'
* isEmpty
 - Returns: boolean
 - Function: Checks if these requirements are empty.'
* setValidDocCodes
```

```
 - Returns: void
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
```

```
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
helper
```

```
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isirRecord
Purpose: The ISIR being evaluated.
Type: IIsirRecordAPI
Available Methods:
* getActiveIsirForAwardYear
 - Returns: Boolean
 - Function: Gets the active ISIR for award year status.'
* getAllFields
 - Returns: @NonNull Map<IsirField<?>,IIsirFieldValue<?>>
 - Function: Retrieves all field values (WARNING: can only be used with 2025+ ISIRs).'
* getAssumptions
 - Returns: List<IAssumptionAPI>
 - Function: Get the assumptions.'
* getAwardYear
 - Returns: String
 - Function: Gets the award year.'
* getCommentCodes
 - Returns: List<ICommentCodeAPI>
 - Function: Get the comment codes.'
* getCpsTransNumber
 - Returns: String
 - Function: Gets the CPS transaction number.'
* getCurrentSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's current SSN.'
* getDiscrepancies
 - Returns: List<IDiscrepancyAPI>
 - Function: Get the discrepancies.'
* getField
 - Returns: <T> @NonNull IIsirFieldValue<T>
```

```
 - Function: Retrieves the value of a given field (WARNING: can only be used with 2025+ ISIRs).'
* getFirstName
 - Returns: String
 - Function: Gets the student's first name.'
* getHighlightFlags
 - Returns: List<IHighlightFlagAPI>
 - Function: Get the highlight flags.'
* getIsirFieldValue
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValueOrNull
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValues
 - Returns: Map<String,String>
 - Function: Get a map containing all ISIR fields (WARNING: This method can only be used up to 2024 award
 year).'
* getIsirIdOriginalNameId
 - Returns: String
 - Function: Gets the student's original name ID.'
* getIsirIdSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's original SSN.'
* getIsirIdYearIndicator
 - Returns: String
 - Function: Gets the ISIR's year indicator.'
* getIsirStatus
 - Returns: String
 - Function: Gets the ISIR status.'
* getIsirStatusBeforeCorrection
 - Returns: String
 - Function: When ISIR Status is Corrected returns the previous ISIR Status.'
* getLastName
 - Returns: String
 - Function: Gets the student's last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the student's middle name.'
* getReceivedDate
 - Returns: Date
 - Function: Gets the system receipt date of the ISIR.'
* getRejectCodes
 - Returns: List<IRejectCodeAPI>
 - Function: Get the reject codes.'
* getSchoolCode1
 - Returns: String
 - Function: Gets the school code 1.'
* getSchoolCode10
 - Returns: String
 - Function: Gets the school code 10.'
* getSchoolCode2
 - Returns: String
 - Function: Gets the school code 2.'
* getSchoolCode3
 - Returns: String
 - Function: Gets the school code 3.'
* getSchoolCode4
 - Returns: String
 - Function: Gets the school code 4.'
* getSchoolCode5
 - Returns: String
 - Function: Gets the school code 5.'
* getSchoolCode6
 - Returns: String
 - Function: Gets the school code 6.'
* getSchoolCode7
```

```
 - Returns: String
 - Function: Gets the school code 7.'
* getSchoolCode8
 - Returns: String
 - Function: Gets the school code 8.'
* getSchoolCode9
 - Returns: String
 - Function: Gets the school code 9.'
* isCompatibleWithPackaging
 - Returns: boolean
 - Function: Checks if this ISIR satisfies minimal acceptance criteria for automatic Financial
Aid Package Calculation.'
* isUnverified
 - Returns: boolean
 - Function: Gets the ISIR verification status.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
nsldsGuarantorServicers
Purpose: List of all guarantors and servicers.
Type: IGuarantorServicersListAPI
Available Methods:
* getAwardYear
 - Returns: String
 - Function: No description available'
* getGuarantorServicerName
 - Returns: String
 - Function: No description available'
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
```

```
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
```

```
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
```

```
    - Returns: IR2t4API
    - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
    predicate.'
  * filterProcessStatuses
    - Returns: IR2t4API
    - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
    given predicate.'
  * getGrossAmountDecreased
    - Returns: BigDecimal
    - Function: No description available'
  * getNetAmountDecreased
    - Returns: BigDecimal
    - Function: No description available'
  * getNullableOldas
    - Returns: ISet<LocalDate>
    - Function: No description available'
  * getNullableProcessStatuses
    - Returns: ISet<String>
    - Function: No description available'
  * getOldas
    - Returns: ISet<LocalDate>
    - Function: No description available'
  * getProcessStatuses
    - Returns: ISet<String>
    - Function: No description available'
  * getWithLatest
    - Returns: IR2t4API
    - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
  * getWithOlda
    - Returns: IR2t4API
    - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
  * getWithOldas
    - Returns: IR2t4API
    - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
  * getWithProcessStatus
    - Returns: IR2t4API
    - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
  * getWithProcessStatuses
    - Returns: IR2t4API
    - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
  * isEligibleForPWD
    - Returns: boolean
    - Function: No description available'
  * isLatest
    - Returns: boolean
    - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Active documents for the student for the award year of the ISIR being evaluated.
Type: IReceivedDocs
Available Methods:
  * get
    - Returns: IReceivedDoc
    - Function: Return doc for specified code and Student, if it exists.'
  * get
    - Returns: IReceivedDoc
    - Function: Return doc for specified code and owners, if it exists.'
  * get
    - Returns: IReceivedDoc
    - Function: Return doc for specified code, owners, and subtype, if it exists.'
  * getDocumentsForDocCode
    - Returns: List<IReceivedDoc>
    - Function: Return list with doc or doc subtpyes for specified code and Student, if it exists.'
  * getDocumentsForDocCode
    - Returns: List<IReceivedDoc>
    - Function: Return list with doc or doc subtpyes for specified code and owners, if it exists.'
  * getValidDocCodes
```

```
 - Returns: Set<String>
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: Return true if doc exists with the right code and owners'
* noDocs
 - Returns: boolean
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
recentFahRecords
Purpose: All Financial Aid History records associated with the student.
Type: IRecentFahRecordsAPI
Available Methods:
* hasRecordsOfType
 - Returns: boolean
 - Function: No description available'
* overlapsLoan
 - Returns: boolean
 - Function: No description available'
* getRecords
 - Returns: List<IFahRecordAPI>
 - Function: No description available'
* getRecordsByType
 - Returns: List<IFahRecordAPI>
 - Function: No description available'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
```

```
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
throwsDocumentIsFaUnacceptableExceptionParam
Purpose: This binding can be set within a Groovy script to control whether a subsequent call to
IReceivedDoc.isAcceptable() or IReceivedDocDocumentAPI.isAcceptable() triggers an exception or not when a
document is unnaceptable. The default value is true.
Type: Boolean
Available Methods:
No methods available
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
```

```
   - Critical business rules enforced
   - Boundary conditions and limits
   - Financial aid implications
   - Required inputs and expected outputs
   - Error handling approach
   - Integration points
   - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
   Format:
   // CONDITIONALS:
   // Check [number]: [specific condition being evaluated]
   // Variables: [list of variables and methods involved]
   // Purpose: [detailed business reason for this check]
   // Logic:
   // - [step-by-step breakdown of the condition]
   // - [how variables are used]
   // - [any transformations applied]
   // Success Path: [what happens when condition is true]
   // Failure Path: [what happens when condition is false]
   // Business Rules: [specific rules being enforced]
   // Error Handling: [how errors are managed]
   // Financial Impact: [effect on aid calculations]
   // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
   Format:
   // BOUNDARY CHECKS:
   // Check [number]: [specific limit being enforced]
   // Source: [where boundary values come from]
   // Variables: [variables and methods involved]
   // Validation:
   // - [how boundaries are checked]
   // - [validation method details]
   // Edge Cases:
   // - [specific edge cases handled]
   // - [how each is managed]
   // Error Handling: [boundary violation management]
   // Impact: [effect of boundary violations]
   // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
   Format:
   // ARITHMETIC:
   // Operation [number]: [name of calculation]
   // Formula: [complete mathematical expression]
   // Variables:
   // - [each variable and its purpose]
   // - [data types and units]
   // Steps:
   // 1. [detailed step-by-step breakdown]
   // 2. [intermediate calculations]
   // 3. [transformations applied]
   // Business Context: [why this calculation is needed]
   // Result: [meaning of the output]
   // Financial Impact: [effect on aid calculations]
   // Limitations: [any mathematical constraints]
   // If N/A, explicitly state: "ARITHMETIC: N/A"
ADDITIONAL REQUIREMENTS:
1. Complex Logic Sections:
   - Must break down multi-step operations
   - Explain interactions between components
   - Detail data flow and transformations
2. Error Handling:
   - Document all error conditions
   - Explain recovery procedures
   - Detail logging and reporting
```

```
3. API Interactions:
 - Document all platform method calls
 - Explain data transformations
 - Detail error handling
4. Financial Aid Impact:
 - Explain effects on aid calculations
 - Document regulatory compliance
 - Detail student eligibility impact
Example Section With All Aspects:
// CONDITIONALS:
// Check 1: Payment Period Status Validation
// Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
N/A - There is no baseline config. This is a placeholder.
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
```

```
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# ISIR Reject Code Documents Required

Name: isirRejectCodeDocumentsRequired

Purpose: Determine the documents required to resolve an ISIR reject code.

In Baseline Config: No

Return Type: Void

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "isirRejectCodeDocumentsRequired"
Primary Function: Determine documents required to resolve an ISIR reject code.
Expected Output Type: Void
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
AGGREGATES_PERKINS_FLAGS
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
GRANT_PAYMENT
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
LOAN_DETAIL
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
OVERPAYMENT
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
allAwardYearReceivedDocuments
Purpose: Provides access to all documents received for the student.
```

```
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
```

```
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
```

```
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
documentRequest
Purpose: Binding that shall be populated by the script with the selected documents.
Type: IIsirRequiredDocsAPI
Available Methods:
* addAnyDocuments
 - Returns: void
 - Function: Adds a list of documents, any one in the list is required.'
* addAnyDocuments
 - Returns: void
 - Function: Adds a list of documents, any one in the list is required.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* any
 - Returns: IIsirRequiredDocsAPI
 - Function: No description available'
* clear
 - Returns: void
 - Function: Clears all docs.'
* doc
 - Returns: IIsirRequiredDocsAPI
 - Function: No description available'
* getAdditionalDocRequestMessages
 - Returns: List<String>
 - Function: No description available'
* getValidDocCodes
 - Returns: Set<String>
 - Function: No description available'
* hasDocument
 - Returns: boolean
 - Function: Returns true if the document has been added, either as a regular request
or an 'any' request.'
* hasDocument
 - Returns: boolean
 - Function: Returns true if the document has been added, either as a regular request
or an 'any' request.'
* isEmpty
 - Returns: boolean
 - Function: Checks if these requirements are empty.'
* setValidDocCodes
 - Returns: void
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
```

```
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
```

```
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
```

```
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isirRecord
Purpose: The ISIR being evaluated.
Type: IIsirRecordAPI
Available Methods:
* getActiveIsirForAwardYear
 - Returns: Boolean
 - Function: Gets the active ISIR for award year status.'
* getAllFields
 - Returns: @NonNull Map<IsirField<?>,IIsirFieldValue<?>>
 - Function: Retrieves all field values (WARNING: can only be used with 2025+ ISIRs).'
* getAssumptions
 - Returns: List<IAssumptionAPI>
 - Function: Get the assumptions.'
* getAwardYear
 - Returns: String
 - Function: Gets the award year.'
* getCommentCodes
 - Returns: List<ICommentCodeAPI>
 - Function: Get the comment codes.'
* getCpsTransNumber
 - Returns: String
 - Function: Gets the CPS transaction number.'
* getCurrentSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's current SSN.'
* getDiscrepancies
 - Returns: List<IDiscrepancyAPI>
 - Function: Get the discrepancies.'
* getField
 - Returns: <T> @NonNull IIsirFieldValue<T>
 - Function: Retrieves the value of a given field (WARNING: can only be used with 2025+ ISIRs).'
* getFirstName
 - Returns: String
 - Function: Gets the student's first name.'
* getHighlightFlags
 - Returns: List<IHighlightFlagAPI>
 - Function: Get the highlight flags.'
* getIsirFieldValue
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValueOrNull
```

```
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValues
 - Returns: Map<String,String>
 - Function: Get a map containing all ISIR fields (WARNING: This method can only be used up to 2024 award
 year).'
* getIsirIdOriginalNameId
 - Returns: String
 - Function: Gets the student's original name ID.'
* getIsirIdSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's original SSN.'
* getIsirIdYearIndicator
 - Returns: String
 - Function: Gets the ISIR's year indicator.'
* getIsirStatus
 - Returns: String
 - Function: Gets the ISIR status.'
* getIsirStatusBeforeCorrection
 - Returns: String
 - Function: When ISIR Status is Corrected returns the previous ISIR Status.'
* getLastName
 - Returns: String
 - Function: Gets the student's last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the student's middle name.'
* getReceivedDate
 - Returns: Date
 - Function: Gets the system receipt date of the ISIR.'
* getRejectCodes
 - Returns: List<IRejectCodeAPI>
 - Function: Get the reject codes.'
* getSchoolCode1
 - Returns: String
 - Function: Gets the school code 1.'
* getSchoolCode10
 - Returns: String
 - Function: Gets the school code 10.'
* getSchoolCode2
 - Returns: String
 - Function: Gets the school code 2.'
* getSchoolCode3
 - Returns: String
 - Function: Gets the school code 3.'
* getSchoolCode4
 - Returns: String
 - Function: Gets the school code 4.'
* getSchoolCode5
 - Returns: String
 - Function: Gets the school code 5.'
* getSchoolCode6
 - Returns: String
 - Function: Gets the school code 6.'
* getSchoolCode7
 - Returns: String
 - Function: Gets the school code 7.'
* getSchoolCode8
 - Returns: String
 - Function: Gets the school code 8.'
* getSchoolCode9
 - Returns: String
 - Function: Gets the school code 9.'
* isCompatibleWithPackaging
 - Returns: boolean
 - Function: Checks if this ISIR satisfies minimal acceptance criteria for automatic Financial
```

```
Aid Package Calculation.'
* isUnverified
 - Returns: boolean
 - Function: Gets the ISIR verification status.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
nsldsGuarantorServicers
Purpose: List of all guarantors and servicers.
Type: IGuarantorServicersListAPI
Available Methods:
* getAwardYear
 - Returns: String
 - Function: No description available'
* getGuarantorServicerName
 - Returns: String
 - Function: No description available'
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
```

```
    - Returns: String
    - Function: No description available'
  * getAssessedUnits
    - Returns: BigDecimal
    - Function: No description available'
  * getAydUnits
    - Returns: BigDecimal
    - Function: No description available'
  * getAydWeeks
    - Returns: Integer
    - Function: No description available'
  * getCharges
    - Returns: List<IChargeAPI>
    - Function: No description available'
  * getCode
    - Returns: String
    - Function: No description available'
  * getCollege
    - Returns: String
    - Function: No description available'
  * getCourses
    - Returns: ICoursesAPI
    - Function: No description available'
  * getDateOfDetermination
    - Returns: LocalDate
    - Function: No description available'
  * getDescription
    - Returns: String
    - Function: No description available'
  * getEnrollmentStatus
    - Returns: String
    - Function: No description available'
  * getEnrollmentStatusEffectiveDate
    - Returns: LocalDate
    - Function: No description available'
  * getEnrollmentStatusSubType
    - Returns: String
    - Function: No description available'
  * getExternalProgramId
    - Returns: String
    - Function: No description available'
  * getFaEligibleIndicator
    - Returns: Boolean
    - Function: No description available'
  * getGpa
    - Returns: BigDecimal
    - Function: No description available'
  * getManualSapEvaluationIndicator
    - Returns: Boolean
    - Function: No description available'
  * getModality
    - Returns: String
    - Function: No description available'
  * getOfficialLastDateOfAttendance
    - Returns: LocalDate
    - Function: No description available'
  * getOpeId
    - Returns: String
    - Function: No description available'
  * getPrimaryLocation
    - Returns: String
    - Function: No description available'
  * getProgramLengthMeasurementUnit
    - Returns: String
    - Function: No description available'
  * getProgramLengthNumberOfUnits
```

```
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
```

```
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Active documents for the student for the award year of the ISIR being evaluated.
Type: IReceivedDocs
Available Methods:
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and Student, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and owners, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code, owners, and subtype, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and Student, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and owners, if it exists.'
* getValidDocCodes
 - Returns: Set<String>
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: Return true if doc exists with the right code and owners'
* noDocs
 - Returns: boolean
 - Function: No description available'
```

```
* put
 - Returns: void
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
recentFahRecords
Purpose: All Financial Aid History records associated with the student.
Type: IRecentFahRecordsAPI
Available Methods:
* hasRecordsOfType
 - Returns: boolean
 - Function: No description available'
* overlapsLoan
 - Returns: boolean
 - Function: No description available'
* getRecords
 - Returns: List<IFahRecordAPI>
 - Function: No description available'
* getRecordsByType
 - Returns: List<IFahRecordAPI>
 - Function: No description available'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
```

```
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
throwsDocumentIsFaUnacceptableExceptionParam
Purpose: This binding can be set within a Groovy script to control whether a subsequent call to
IReceivedDoc.isAcceptable() or IReceivedDocDocumentAPI.isAcceptable() triggers an exception or not when a
document is unnaceptable. The default value is true.
Type: Boolean
Available Methods:
No methods available
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
```

```
    // CONDITIONALS:
    // Check [number]: [specific condition being evaluated]
    // Variables: [list of variables and methods involved]
    // Purpose: [detailed business reason for this check]
    // Logic:
    // - [step-by-step breakdown of the condition]
    // - [how variables are used]
    // - [any transformations applied]
    // Success Path: [what happens when condition is true]
    // Failure Path: [what happens when condition is false]
    // Business Rules: [specific rules being enforced]
    // Error Handling: [how errors are managed]
    // Financial Impact: [effect on aid calculations]
    // If N/A, explicitly state: "CONDITIONALS: N/A"
  2. BOUNDARY CHECKS Section:
    Format:
    // BOUNDARY CHECKS:
    // Check [number]: [specific limit being enforced]
    // Source: [where boundary values come from]
    // Variables: [variables and methods involved]
    // Validation:
    // - [how boundaries are checked]
    // - [validation method details]
    // Edge Cases:
    // - [specific edge cases handled]
    // - [how each is managed]
    // Error Handling: [boundary violation management]
    // Impact: [effect of boundary violations]
    // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
  3. ARITHMETIC Section:
    Format:
    // ARITHMETIC:
    // Operation [number]: [name of calculation]
    // Formula: [complete mathematical expression]
    // Variables:
    // - [each variable and its purpose]
    // - [data types and units]
    // Steps:
    // 1. [detailed step-by-step breakdown]
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
  ADDITIONAL REQUIREMENTS:
  1. Complex Logic Sections:
    - Must break down multi-step operations
    - Explain interactions between components
    - Detail data flow and transformations
  2. Error Handling:
    - Document all error conditions
    - Explain recovery procedures
    - Detail logging and reporting
  3. API Interactions:
    - Document all platform method calls
    - Explain data transformations
    - Detail error handling
  4. Financial Aid Impact:
    - Explain effects on aid calculations
    - Document regulatory compliance
    - Detail student eligibility impact
  Example Section With All Aspects:
  // CONDITIONALS:
  // Check 1: Payment Period Status Validation
```

```
// Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
N/A - There is no baseline config. This is a placeholder.
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# ISIR Verification Documents Required

Name: isirVerificationDocumentsRequired

Purpose: Determine the documents required to resolve an ISIR verification.

In Baseline Config: Yes

Return Type: Void

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "isirVerificationDocumentsRequired"
Primary Function: Determine documents required to resolve an ISIR verification.
Expected Output Type: Void
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
isirRecord
Purpose: The ISIR being evaluated.
Type: IIsirRecordAPI
Available Methods:
* getActiveIsirForAwardYear
 - Returns: Boolean
 - Function: Gets the active ISIR for award year status.'
* getAllFields
 - Returns: @NonNull Map<IsirField<?>,IIsirFieldValue<?>>
 - Function: Retrieves all field values (WARNING: can only be used with 2025+ ISIRs).'
* getAssumptions
 - Returns: List<IAssumptionAPI>
 - Function: Get the assumptions.'
* getAwardYear
 - Returns: String
 - Function: Gets the award year.'
* getCommentCodes
 - Returns: List<ICommentCodeAPI>
 - Function: Get the comment codes.'
* getCpsTransNumber
 - Returns: String
 - Function: Gets the CPS transaction number.'
* getCurrentSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's current SSN.'
* getDiscrepancies
 - Returns: List<IDiscrepancyAPI>
 - Function: Get the discrepancies.'
* getField
 - Returns: <T> @NonNull IIsirFieldValue<T>
 - Function: Retrieves the value of a given field (WARNING: can only be used with 2025+ ISIRs).'
```

```
* getFirstName
 - Returns: String
 - Function: Gets the student's first name.'
* getHighlightFlags
 - Returns: List<IHighlightFlagAPI>
 - Function: Get the highlight flags.'
* getIsirFieldValue
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValueOrNull
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValues
 - Returns: Map<String,String>
 - Function: Get a map containing all ISIR fields (WARNING: This method can only be used up to 2024 award
 year).'
* getIsirIdOriginalNameId
 - Returns: String
 - Function: Gets the student's original name ID.'
* getIsirIdSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's original SSN.'
* getIsirIdYearIndicator
 - Returns: String
 - Function: Gets the ISIR's year indicator.'
* getIsirStatus
 - Returns: String
 - Function: Gets the ISIR status.'
* getIsirStatusBeforeCorrection
 - Returns: String
 - Function: When ISIR Status is Corrected returns the previous ISIR Status.'
* getLastName
 - Returns: String
 - Function: Gets the student's last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the student's middle name.'
* getReceivedDate
 - Returns: Date
 - Function: Gets the system receipt date of the ISIR.'
* getRejectCodes
 - Returns: List<IRejectCodeAPI>
 - Function: Get the reject codes.'
* getSchoolCode1
 - Returns: String
 - Function: Gets the school code 1.'
* getSchoolCode10
 - Returns: String
 - Function: Gets the school code 10.'
* getSchoolCode2
 - Returns: String
 - Function: Gets the school code 2.'
* getSchoolCode3
 - Returns: String
 - Function: Gets the school code 3.'
* getSchoolCode4
 - Returns: String
 - Function: Gets the school code 4.'
* getSchoolCode5
 - Returns: String
 - Function: Gets the school code 5.'
* getSchoolCode6
 - Returns: String
 - Function: Gets the school code 6.'
* getSchoolCode7
 - Returns: String
```

```
 - Function: Gets the school code 7.'
* getSchoolCode8
 - Returns: String
 - Function: Gets the school code 8.'
* getSchoolCode9
 - Returns: String
 - Function: Gets the school code 9.'
* isCompatibleWithPackaging
 - Returns: boolean
 - Function: Checks if this ISIR satisfies minimal acceptance criteria for automatic Financial
Aid Package Calculation.'
* isUnverified
 - Returns: boolean
 - Function: Gets the ISIR verification status.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
nsldsGuarantorServicers
Purpose: List of all guarantors and servicers.
Type: IGuarantorServicersListAPI
Available Methods:
* getAwardYear
 - Returns: String
 - Function: No description available'
* getGuarantorServicerName
 - Returns: String
 - Function: No description available'
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
```

```
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
```

```
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
```

```
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Active documents for the student for the award year of the ISIR being evaluated.
Type: IReceivedDocs
Available Methods:
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and Student, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and owners, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code, owners, and subtype, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and Student, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and owners, if it exists.'
* getValidDocCodes
 - Returns: Set<String>
```

```
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: Return true if doc exists with the right code and owners'
* noDocs
 - Returns: boolean
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
recentFahRecords
Purpose: All Financial Aid History records associated with the student.
Type: IRecentFahRecordsAPI
Available Methods:
* hasRecordsOfType
 - Returns: boolean
 - Function: No description available'
* overlapsLoan
 - Returns: boolean
 - Function: No description available'
* getRecords
 - Returns: List<IFahRecordAPI>
 - Function: No description available'
* getRecordsByType
 - Returns: List<IFahRecordAPI>
 - Function: No description available'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
```

```
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
throwsDocumentIsFaUnacceptableExceptionParam
Purpose: This binding can be set within a Groovy script to control whether a subsequent call to
IReceivedDoc.isAcceptable() or IReceivedDocDocumentAPI.isAcceptable() triggers an exception or not when a
document is unnaceptable. The default value is true.
Type: Boolean
Available Methods:
No methods available
AGGREGATES_PERKINS_FLAGS
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
GRANT_PAYMENT
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
LOAN_DETAIL
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
```

```
No methods available
OVERPAYMENT
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
allAwardYearReceivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
```

```
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
```

```
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
documentRequest
Purpose: Binding that shall be populated by the script with the selected documents.
Type: IIsirRequiredDocsAPI
Available Methods:
* addAnyDocuments
 - Returns: void
 - Function: Adds a list of documents, any one in the list is required.'
* addAnyDocuments
 - Returns: void
 - Function: Adds a list of documents, any one in the list is required.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* any
 - Returns: IIsirRequiredDocsAPI
 - Function: No description available'
* clear
 - Returns: void
 - Function: Clears all docs.'
* doc
 - Returns: IIsirRequiredDocsAPI
 - Function: No description available'
* getAdditionalDocRequestMessages
 - Returns: List<String>
 - Function: No description available'
* getValidDocCodes
 - Returns: Set<String>
 - Function: No description available'
* hasDocument
 - Returns: boolean
 - Function: Returns true if the document has been added, either as a regular request
or an 'any' request.'
* hasDocument
 - Returns: boolean
 - Function: Returns true if the document has been added, either as a regular request
or an 'any' request.'
* isEmpty
 - Returns: boolean
 - Function: Checks if these requirements are empty.'
* setValidDocCodes
 - Returns: void
 - Function: No description available'
enrollment
```

```
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
```

```
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
```

```
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
```

```
    // Failure Path: [what happens when condition is false]
    // Business Rules: [specific rules being enforced]
    // Error Handling: [how errors are managed]
    // Financial Impact: [effect on aid calculations]
    // If N/A, explicitly state: "CONDITIONALS: N/A"
   2. BOUNDARY CHECKS Section:
    Format:
    // BOUNDARY CHECKS:
    // Check [number]: [specific limit being enforced]
    // Source: [where boundary values come from]
    // Variables: [variables and methods involved]
    // Validation:
    // - [how boundaries are checked]
    // - [validation method details]
    // Edge Cases:
    // - [specific edge cases handled]
    // - [how each is managed]
    // Error Handling: [boundary violation management]
    // Impact: [effect of boundary violations]
    // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
   3. ARITHMETIC Section:
    Format:
    // ARITHMETIC:
    // Operation [number]: [name of calculation]
    // Formula: [complete mathematical expression]
    // Variables:
    // - [each variable and its purpose]
    // - [data types and units]
    // Steps:
    // 1. [detailed step-by-step breakdown]
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
   ADDITIONAL REQUIREMENTS:
   1. Complex Logic Sections:
    - Must break down multi-step operations
    - Explain interactions between components
    - Detail data flow and transformations
   2. Error Handling:
    - Document all error conditions
    - Explain recovery procedures
    - Detail logging and reporting
   3. API Interactions:
    - Document all platform method calls
    - Explain data transformations
    - Detail error handling
   4. Financial Aid Impact:
    - Explain effects on aid calculations
    - Document regulatory compliance
    - Detail student eligibility impact
   Example Section With All Aspects:
   // CONDITIONALS:
   // Check 1: Payment Period Status Validation
   // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
   // Purpose: Ensure only valid, active payment periods are included
   // Logic:
   // - Retrieve all loan payment periods
   // - Filter based on status (!= "CANCELED")
   // - Compare start dates with reference date
   // Success Path: Period included in calculations
   // Failure Path: Period excluded from totals
   // Business Rules: Canceled periods must not affect grade level
```

```
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.IsirVerificationDocumentsRequiredScript)
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
//Student ISIR variables
def isirSTUDENT_FILED1040OR1040NR = isirRecord.getField(IsirField.STUDENT_FILED1040OR1040NR).getOrNull()
def isirSTUDENTFTIM_IRSRESPONSECODE_FTI =
 isirRecord.getField(IsirField.STUDENTFTIM_IRSRESPONSECODE_FTI).getOrNull()
def isirSTUDENT_FILEDNONUSTAXRETURN = isirRecord.getField(IsirField.STUDENT_FILEDNONUSTAXRETURN).getOrNull()
def isirSTUDENTFTIM_UNTAXEDIRADISTRIBUTIONS_FTI =
 isirRecord.getField(IsirField.STUDENTFTIM_UNTAXEDIRADISTRIBUTIONS_FTI).getOrNull()
def isirSTUDENTFTIM_IRADEDUCTIBLEANDPAYMENTS_FTI =
 isirRecord.getField(IsirField.STUDENTFTIM_IRADEDUCTIBLEANDPAYMENTS_FTI).getOrNull()
def isirSTUDENT_UNTAXEDPORTIONSOFIRADISTRIBUTIONS =
 isirRecord.getField(IsirField.STUDENT_UNTAXEDPORTIONSOFIRADISTRIBUTIONS).getOrNull()
def isirSTUDENT_IRAROLLOVER = isirRecord.getField(IsirField.STUDENT_IRAROLLOVER).getOrNull()
def isirSTUDENT_UPDATEDFAMILYSIZE = isirRecord.getField(IsirField.STUDENT_UPDATEDFAMILYSIZE).getOrNull()
def isirFPS_INCARCERATEDAPPLICANTFLAG =
 isirRecord.getField(IsirField.FPS_INCARCERATEDAPPLICANTFLAG).getOrNull()
def isirFPS_ASSUMEDSTUDENTFAMILYSIZE =
 isirRecord.getField(IsirField.FPS_ASSUMEDSTUDENTFAMILYSIZE).getOrNull()
def isirSTUDENT_FOREIGNEARNEDINCOMEEXCLUSION =
 isirRecord.getField(IsirField.STUDENT_FOREIGNEARNEDINCOMEEXCLUSION).getOrNull()
//Student Spouse ISIR variables
def isirSPOUSEFTIM_IRSRESPONSECODE_FTI =
 isirRecord.getField(IsirField.SPOUSEFTIM_IRSRESPONSECODE_FTI).getOrNull()
def isirSPOUSE_TAXRETURNFILINGSTATUS =
 isirRecord.getField(IsirField.SPOUSE_TAXRETURNFILINGSTATUS).getOrNull()
def isirSPOUSEFTIM_UNTAXEDIRADISTRIBUTIONS_FTI =
 isirRecord.getField(IsirField.SPOUSEFTIM_UNTAXEDIRADISTRIBUTIONS_FTI).getOrNull()
def isirSPOUSEFTIM_IRADEDUCTIBLEANDPAYMENTS_FTI =
 isirRecord.getField(IsirField.SPOUSEFTIM_IRADEDUCTIBLEANDPAYMENTS_FTI).getOrNull()
```
```

```
def isirSPOUSE_UNTAXEDPORTIONSOFIRADISTRIBUTIONS =
 isirRecord.getField(IsirField.SPOUSE_UNTAXEDPORTIONSOFIRADISTRIBUTIONS).getOrNull()
def isirSPOUSE_IRAROLLOVER = isirRecord.getField(IsirField.SPOUSE_IRAROLLOVER).getOrNull()
def isirSPOUSE_FOREIGNEARNEDINCOMEEXCLUSION =
 isirRecord.getField(IsirField.SPOUSE_FOREIGNEARNEDINCOMEEXCLUSION).getOrNull()
//If student is incarcerated then exit the script
if (isirFPS_INCARCERATEDAPPLICANTFLAG == "1" || isirFPS_INCARCERATEDAPPLICANTFLAG == "2" ||
 isirFPS_INCARCERATEDAPPLICANTFLAG == "3") {
 return
}
//Student document request logic
if ((isirSTUDENT_FILED1040OR1040NR == "1" || isirSTUDENT_FILED1040OR1040NR == null) && [null, "203", "206",
 "212", "214"].contains(isirSTUDENTFTIM_IRSRESPONSECODE_FTI)) {
 documentRequest.addAnyDocuments(["1040", "Student"], ["1040x", "Student"], ["TaxReturnTranscript",
 "Student"])
}
else if (isirSTUDENT_FILED1040OR1040NR == "2" && isirSTUDENT_FILEDNONUSTAXRETURN == "1") {
 documentRequest.addDocument("ForeignTaxTranscript", "Student")
}
else if (isirSTUDENT_FILED1040OR1040NR == "2" && (isirSTUDENT_FILEDNONUSTAXRETURN == "2" ||
 isirSTUDENT_FILEDNONUSTAXRETURN == null)) {
 documentRequest.addAnyDocuments(["NonFilingStatement", "Student"], ["W2", "Student"], ["1099G", "Student"],
 ["SelfEmploymentStatement", "Student"])
}
//Student Spouse document request logic
if (isirSPOUSE_TAXRETURNFILINGSTATUS == "3" && [null, "203", "206", "212",
 "214"].contains(isirSPOUSEFTIM_IRSRESPONSECODE_FTI)) {
 documentRequest.addAnyDocuments(["1040", "Student Spouse"], ["1040x", "Student Spouse"],
 ["TaxReturnTranscript", "Student Spouse"])
}
else if (isirSPOUSE_TAXRETURNFILINGSTATUS == "3" && [null, "203", "206", "212",
 "214"].contains(isirSPOUSEFTIM_IRSRESPONSECODE_FTI) && isirSTUDENTFTIM_IRSRESPONSECODE_FTI == "200") {
 documentRequest.addAnyDocuments(["1040", "Student Spouse"], ["1040x", "Student Spouse"],
 ["TaxReturnTranscript", "Student Spouse"])
}
//Student rollover information document request logic
if (isirSTUDENTFTIM_UNTAXEDIRADISTRIBUTIONS_FTI > 0 || isirSTUDENTFTIM_IRADEDUCTIBLEANDPAYMENTS_FTI > 0 ||
 isirSTUDENT_UNTAXEDPORTIONSOFIRADISTRIBUTIONS > 0 || isirSTUDENT_IRAROLLOVER > 0) {
 documentRequest.addDocument("RolloverStatement", "Student")
}
//Student Spouse rollover information document request logic
if (isirSPOUSEFTIM_UNTAXEDIRADISTRIBUTIONS_FTI > 0 || isirSPOUSEFTIM_IRADEDUCTIBLEANDPAYMENTS_FTI > 0 ||
 isirSPOUSE_UNTAXEDPORTIONSOFIRADISTRIBUTIONS > 0 || isirSPOUSE_IRAROLLOVER > 0) {
 documentRequest.addDocument("RolloverStatement", "Student Spouse")
}
//Student Foreign Income Exclusion document request logic
if (isirSTUDENT_FOREIGNEARNEDINCOMEEXCLUSION != null && (isirSTUDENT_FOREIGNEARNEDINCOMEEXCLUSION > 0 ||
 isirSTUDENT_FOREIGNEARNEDINCOMEEXCLUSION < 0)) {
 documentRequest.addAnyDocuments(["1040", "Student"], ["1040x", "Student"], ["TaxReturnTranscript",
 "Student"], ["ForeignTaxTranscript", "Student"])
}
//Student Spouse Foreign Income Exclusion document request logic
if (isirSPOUSE_FOREIGNEARNEDINCOMEEXCLUSION != null && (isirSPOUSE_FOREIGNEARNEDINCOMEEXCLUSION > 0 ||
 isirSPOUSE_FOREIGNEARNEDINCOMEEXCLUSION < 0)) {
 documentRequest.addAnyDocuments(["1040", "Student Spouse"], ["1040x", "Student Spouse"],
 ["TaxReturnTranscript", "Student Spouse"], ["ForeignTaxTranscript", "Student Spouse"])
}
//Family size document request logic
if (isirSTUDENT_UPDATEDFAMILYSIZE != null || isirFPS_ASSUMEDSTUDENTFAMILYSIZE != null) {
 documentRequest.addDocument("VW-Ind", "Student")
}
return
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:

```
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# ISIR Assumption Documents Required

Name: isirAssumptionDocumentsRequired

Purpose: Determine the documents required to resolve an ISIR assumption.

In Baseline Config: No

Return Type: Void

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "isirAssumptionDocumentsRequired"
Primary Function: Determine documents required to resolve an ISIR assumption.
Expected Output Type: Void
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
AGGREGATES_PERKINS_FLAGS
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
GRANT_PAYMENT
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
LOAN_DETAIL
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
```

```
Available Methods:
No methods available
OVERPAYMENT
Purpose: Constant value for the FahRecordType enum.
Type: FahRecordType
Available Methods:
No methods available
allAwardYearReceivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
```

```
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
```

```
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
documentRequest
Purpose: Binding that shall be populated by the script with the selected documents.
Type: IIsirRequiredDocsAPI
Available Methods:
* addAnyDocuments
 - Returns: void
 - Function: Adds a list of documents, any one in the list is required.'
* addAnyDocuments
 - Returns: void
 - Function: Adds a list of documents, any one in the list is required.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* any
 - Returns: IIsirRequiredDocsAPI
 - Function: No description available'
* clear
 - Returns: void
 - Function: Clears all docs.'
* doc
 - Returns: IIsirRequiredDocsAPI
 - Function: No description available'
* getAdditionalDocRequestMessages
 - Returns: List<String>
 - Function: No description available'
* getValidDocCodes
 - Returns: Set<String>
 - Function: No description available'
* hasDocument
 - Returns: boolean
 - Function: Returns true if the document has been added, either as a regular request
or an 'any' request.'
* hasDocument
 - Returns: boolean
 - Function: Returns true if the document has been added, either as a regular request
or an 'any' request.'
* isEmpty
 - Returns: boolean
 - Function: Checks if these requirements are empty.'
* setValidDocCodes
 - Returns: void
 - Function: No description available'
```

```
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
```

```
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
```

```
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isirRecord
Purpose: The ISIR being evaluated.
Type: IIsirRecordAPI
Available Methods:
* getActiveIsirForAwardYear
 - Returns: Boolean
 - Function: Gets the active ISIR for award year status.'
* getAllFields
 - Returns: @NonNull Map<IsirField<?>,IIsirFieldValue<?>>
 - Function: Retrieves all field values (WARNING: can only be used with 2025+ ISIRs).'
* getAssumptions
 - Returns: List<IAssumptionAPI>
 - Function: Get the assumptions.'
* getAwardYear
 - Returns: String
 - Function: Gets the award year.'
* getCommentCodes
 - Returns: List<ICommentCodeAPI>
 - Function: Get the comment codes.'
* getCpsTransNumber
 - Returns: String
 - Function: Gets the CPS transaction number.'
* getCurrentSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's current SSN.'
* getDiscrepancies
 - Returns: List<IDiscrepancyAPI>
 - Function: Get the discrepancies.'
* getField
 - Returns: <T> @NonNull IIsirFieldValue<T>
 - Function: Retrieves the value of a given field (WARNING: can only be used with 2025+ ISIRs).'
* getFirstName
```

```
 - Returns: String
 - Function: Gets the student's first name.'
* getHighlightFlags
 - Returns: List<IHighlightFlagAPI>
 - Function: Get the highlight flags.'
* getIsirFieldValue
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValueOrNull
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValues
 - Returns: Map<String,String>
 - Function: Get a map containing all ISIR fields (WARNING: This method can only be used up to 2024 award
 year).'
* getIsirIdOriginalNameId
 - Returns: String
 - Function: Gets the student's original name ID.'
* getIsirIdSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's original SSN.'
* getIsirIdYearIndicator
 - Returns: String
 - Function: Gets the ISIR's year indicator.'
* getIsirStatus
 - Returns: String
 - Function: Gets the ISIR status.'
* getIsirStatusBeforeCorrection
 - Returns: String
 - Function: When ISIR Status is Corrected returns the previous ISIR Status.'
* getLastName
 - Returns: String
 - Function: Gets the student's last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the student's middle name.'
* getReceivedDate
 - Returns: Date
 - Function: Gets the system receipt date of the ISIR.'
* getRejectCodes
 - Returns: List<IRejectCodeAPI>
 - Function: Get the reject codes.'
* getSchoolCode1
 - Returns: String
 - Function: Gets the school code 1.'
* getSchoolCode10
 - Returns: String
 - Function: Gets the school code 10.'
* getSchoolCode2
 - Returns: String
 - Function: Gets the school code 2.'
* getSchoolCode3
 - Returns: String
 - Function: Gets the school code 3.'
* getSchoolCode4
 - Returns: String
 - Function: Gets the school code 4.'
* getSchoolCode5
 - Returns: String
 - Function: Gets the school code 5.'
* getSchoolCode6
 - Returns: String
 - Function: Gets the school code 6.'
* getSchoolCode7
 - Returns: String
 - Function: Gets the school code 7.'
```

```
* getSchoolCode8
 - Returns: String
 - Function: Gets the school code 8.'
* getSchoolCode9
 - Returns: String
 - Function: Gets the school code 9.'
* isCompatibleWithPackaging
 - Returns: boolean
 - Function: Checks if this ISIR satisfies minimal acceptance criteria for automatic Financial
Aid Package Calculation.'
* isUnverified
 - Returns: boolean
 - Function: Gets the ISIR verification status.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
nsldsGuarantorServicers
Purpose: List of all guarantors and servicers.
Type: IGuarantorServicersListAPI
Available Methods:
* getAwardYear
 - Returns: String
 - Function: No description available'
* getGuarantorServicerName
 - Returns: String
 - Function: No description available'
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
```

```
    - Returns: boolean
    - Function: No description available'
  * getAcademicCompletionDate
    - Returns: LocalDate
    - Function: No description available'
  * getAcademicStatus
    - Returns: String
    - Function: No description available'
  * getAdmissionStatus
    - Returns: String
    - Function: No description available'
  * getAssessedUnits
    - Returns: BigDecimal
    - Function: No description available'
  * getAydUnits
    - Returns: BigDecimal
    - Function: No description available'
  * getAydWeeks
    - Returns: Integer
    - Function: No description available'
  * getCharges
    - Returns: List<IChargeAPI>
    - Function: No description available'
  * getCode
    - Returns: String
    - Function: No description available'
  * getCollege
    - Returns: String
    - Function: No description available'
  * getCourses
    - Returns: ICoursesAPI
    - Function: No description available'
  * getDateOfDetermination
    - Returns: LocalDate
    - Function: No description available'
  * getDescription
    - Returns: String
    - Function: No description available'
  * getEnrollmentStatus
    - Returns: String
    - Function: No description available'
  * getEnrollmentStatusEffectiveDate
    - Returns: LocalDate
    - Function: No description available'
  * getEnrollmentStatusSubType
    - Returns: String
    - Function: No description available'
  * getExternalProgramId
    - Returns: String
    - Function: No description available'
  * getFaEligibleIndicator
    - Returns: Boolean
    - Function: No description available'
  * getGpa
    - Returns: BigDecimal
    - Function: No description available'
  * getManualSapEvaluationIndicator
    - Returns: Boolean
    - Function: No description available'
  * getModality
    - Returns: String
    - Function: No description available'
  * getOfficialLastDateOfAttendance
    - Returns: LocalDate
    - Function: No description available'
  * getOpeId
```

```
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
```

```
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Active documents for the student for the award year of the ISIR being evaluated.
Type: IReceivedDocs
Available Methods:
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and Student, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and owners, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code, owners, and subtype, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and Student, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and owners, if it exists.'
* getValidDocCodes
 - Returns: Set<String>
```

```
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: Return true if doc exists with the right code and owners'
* noDocs
 - Returns: boolean
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
recentFahRecords
Purpose: All Financial Aid History records associated with the student.
Type: IRecentFahRecordsAPI
Available Methods:
* hasRecordsOfType
 - Returns: boolean
 - Function: No description available'
* overlapsLoan
 - Returns: boolean
 - Function: No description available'
* getRecords
 - Returns: List<IFahRecordAPI>
 - Function: No description available'
* getRecordsByType
 - Returns: List<IFahRecordAPI>
 - Function: No description available'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
```

```
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
throwsDocumentIsFaUnacceptableExceptionParam
Purpose: This binding can be set within a Groovy script to control whether a subsequent call to
IReceivedDoc.isAcceptable() or IReceivedDocDocumentAPI.isAcceptable() triggers an exception or not when a
document is unnaceptable. The default value is true.
Type: Boolean
Available Methods:
No methods available
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
```

```
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
 // Operation [number]: [name of calculation]
 // Formula: [complete mathematical expression]
 // Variables:
 // - [each variable and its purpose]
 // - [data types and units]
 // Steps:
 // 1. [detailed step-by-step breakdown]
 // 2. [intermediate calculations]
 // 3. [transformations applied]
 // Business Context: [why this calculation is needed]
 // Result: [meaning of the output]
 // Financial Impact: [effect on aid calculations]
 // Limitations: [any mathematical constraints]
 // If N/A, explicitly state: "ARITHMETIC: N/A"
ADDITIONAL REQUIREMENTS:
1. Complex Logic Sections:
 - Must break down multi-step operations
 - Explain interactions between components
 - Detail data flow and transformations
2. Error Handling:
 - Document all error conditions
 - Explain recovery procedures
 - Detail logging and reporting
3. API Interactions:
```

```
   - Document all platform method calls
   - Explain data transformations
   - Detail error handling
  4. Financial Aid Impact:
   - Explain effects on aid calculations
   - Document regulatory compliance
   - Detail student eligibility impact
  Example Section With All Aspects:
  // CONDITIONALS:
  // Check 1: Payment Period Status Validation
  // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
  // Purpose: Ensure only valid, active payment periods are included
  // Logic:
  // - Retrieve all loan payment periods
  // - Filter based on status (!= "CANCELED")
  // - Compare start dates with reference date
  // Success Path: Period included in calculations
  // Failure Path: Period excluded from totals
  // Business Rules: Canceled periods must not affect grade level
  // Error Handling: Invalid periods logged and skipped
  // Financial Impact: Affects available loan amounts
  //
  // BOUNDARY CHECKS:
  // Check 1: Payment Period Date Validation
  // Source: Academic Calendar Configuration
  // Variables: period.getStartDate(), referenceDate
  // Validation:
  // - Compare period start date with reference date
  // - Ensure dates are not null
  // Edge Cases:
  // - Null dates are excluded
  // - Future dates are filtered
  // Error Handling: Invalid dates logged and skipped
  // Impact: Prevents future periods from affecting current calculation
  //
  // ARITHMETIC:
  // Operation 1: Payment Period Units Summation
  // Formula: sum(validPeriods.map(period => period.getUnits()))
  // Variables:
  // - period.getUnits(): Decimal, represents credit hours
  // - validPeriods: Array of filtered payment periods
  // Steps:
  // 1. Filter valid payment periods
  // 2. Extract units from each period
  // 3. Sum all units using reduce
  // Business Context: Calculate total completed units for grade level
  // Result: Total units affecting grade level
  // Financial Impact: Determines loan eligibility amounts
  // Limitations: Cannot exceed program maximum units
  ## Groovy Script for Analysis
  ```groovy
  N/A - There is no baseline config. This is a placeholder.
  ```
  ## COMMENT GENERATION REQUIREMENTS:
  1. Add comprehensive comments following the exact format specified
  2. Every logical section must include all three aspects:
   - CONDITIONALS
   - BOUNDARY CHECKS
   - ARITHMETIC
  3. Mark any non-applicable aspect as "N/A" explicitly
  4. Ensure line-by-line explanation of complex logic
  5. Detail all financial aid implications
  DO NOT modify the code itself. Add comments only.
  REMEMBER:
  - Comments must be exhaustively detailed
  - Every logical section needs all three aspects
```

```
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# ISIR Discrepancy Required Documents

Name: isirDiscrepancyRequiredDocuments

Purpose: Determine whether the isirDiscrepancyEvaluation script should be executed.

In Baseline Config: No

Return Type: Void

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "isirDiscrepancyRequiredDocuments"
Primary Function: Determine whether the isirDiscrepancyEvaluation script should be executed, based on
 whether all required
documents necessary to make the ISIR correction determination already exist on the student record.
Expected Output Type: Void
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
allAwardYearReceivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
```

```
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
```

```
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
```

```
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
```

```
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
```

```
    - Returns: IAwardYearHelperAPI
    - Function: Returns Award Year-specific helper.'
  * getIntegerFromSignedIsirNumber
    - Returns: Integer
    - Function: Parses a number represented in ISIR format to Integer.'
  * getIsirDate
    - Returns: Date
    - Function: No description available'
  * getIsirDateFormatStr
    - Returns: String
    - Function: No description available'
  * getSignedIsirNumberFromInteger
    - Returns: String
    - Function: No description available'
  * isirRecordsDifferForListedFields
    - Returns: boolean
    - Function: No description available'
  * mapRanges
    - Returns: IRangeMapStarterAPI
    - Function: No description available'
  * newAwardYearPeriod
    - Returns: IAwardYearAPI
    - Function: Returns the Award Year having a given end year.'
  * newPeriod
    - Returns: IPeriod
    - Function: Returns the IPeriod for one single day.'
  * newPeriod
    - Returns: IPeriod
    - Function: Returns a IPeriod given its start and end dates.'
  isirRecord
  Purpose: The ISIR being evaluated.
  Type: IIsirRecordAPI
  Available Methods:
  * getActiveIsirForAwardYear
    - Returns: Boolean
    - Function: Gets the active ISIR for award year status.'
  * getAllFields
    - Returns: @NonNull Map<IsirField<?>,IIsirFieldValue<?>>
    - Function: Retrieves all field values (WARNING: can only be used with 2025+ ISIRs).'
  * getAssumptions
    - Returns: List<IAssumptionAPI>
    - Function: Get the assumptions.'
  * getAwardYear
    - Returns: String
    - Function: Gets the award year.'
  * getCommentCodes
    - Returns: List<ICommentCodeAPI>
    - Function: Get the comment codes.'
  * getCpsTransNumber
    - Returns: String
    - Function: Gets the CPS transaction number.'
  * getCurrentSsn
    - Returns: ISocialSecurityNumber
    - Function: Gets the student's current SSN.'
  * getDiscrepancies
    - Returns: List<IDiscrepancyAPI>
    - Function: Get the discrepancies.'
  * getField
    - Returns: <T> @NonNull IIsirFieldValue<T>
    - Function: Retrieves the value of a given field (WARNING: can only be used with 2025+ ISIRs).'
  * getFirstName
    - Returns: String
    - Function: Gets the student's first name.'
  * getHighlightFlags
    - Returns: List<IHighlightFlagAPI>
    - Function: Get the highlight flags.'
```

```
* getIsirFieldValue
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValueOrNull
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValues
 - Returns: Map<String,String>
 - Function: Get a map containing all ISIR fields (WARNING: This method can only be used up to 2024 award
 year).'
* getIsirIdOriginalNameId
 - Returns: String
 - Function: Gets the student's original name ID.'
* getIsirIdSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's original SSN.'
* getIsirIdYearIndicator
 - Returns: String
 - Function: Gets the ISIR's year indicator.'
* getIsirStatus
 - Returns: String
 - Function: Gets the ISIR status.'
* getIsirStatusBeforeCorrection
 - Returns: String
 - Function: When ISIR Status is Corrected returns the previous ISIR Status.'
* getLastName
 - Returns: String
 - Function: Gets the student's last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the student's middle name.'
* getReceivedDate
 - Returns: Date
 - Function: Gets the system receipt date of the ISIR.'
* getRejectCodes
 - Returns: List<IRejectCodeAPI>
 - Function: Get the reject codes.'
* getSchoolCode1
 - Returns: String
 - Function: Gets the school code 1.'
* getSchoolCode10
 - Returns: String
 - Function: Gets the school code 10.'
* getSchoolCode2
 - Returns: String
 - Function: Gets the school code 2.'
* getSchoolCode3
 - Returns: String
 - Function: Gets the school code 3.'
* getSchoolCode4
 - Returns: String
 - Function: Gets the school code 4.'
* getSchoolCode5
 - Returns: String
 - Function: Gets the school code 5.'
* getSchoolCode6
 - Returns: String
 - Function: Gets the school code 6.'
* getSchoolCode7
 - Returns: String
 - Function: Gets the school code 7.'
* getSchoolCode8
 - Returns: String
 - Function: Gets the school code 8.'
* getSchoolCode9
 - Returns: String
```

```
 - Function: Gets the school code 9.'
* isCompatibleWithPackaging
 - Returns: boolean
 - Function: Checks if this ISIR satisfies minimal acceptance criteria for automatic Financial
Aid Package Calculation.'
* isUnverified
 - Returns: boolean
 - Function: Gets the ISIR verification status.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
previousIsirRecords
Purpose: Previous ISIR records with the same award year as the ISIR being evaluated, ordered by descending
 CPS numbers.
Type: List
Available Methods:
No methods available
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
```

```
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Active documents for the student for the award year of the ISIR being evaluated.
Type: IReceivedDocs
Available Methods:
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and Student, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and owners, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code, owners, and subtype, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and Student, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and owners, if it exists.'
* getValidDocCodes
 - Returns: Set<String>
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: Return true if doc exists with the right code and owners'
* noDocs
```

```
 - Returns: boolean
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
requiredDocuments
Purpose: Binding that shall be populated by the script with the required documents.
Type: IIsirRequiredDocsAPI
Available Methods:
* addAnyDocuments
 - Returns: void
 - Function: Adds a list of documents, any one in the list is required.'
* addAnyDocuments
 - Returns: void
 - Function: Adds a list of documents, any one in the list is required.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* any
 - Returns: IIsirRequiredDocsAPI
 - Function: No description available'
* clear
 - Returns: void
 - Function: Clears all docs.'
* doc
 - Returns: IIsirRequiredDocsAPI
 - Function: No description available'
* getAdditionalDocRequestMessages
 - Returns: List<String>
 - Function: No description available'
* getValidDocCodes
 - Returns: Set<String>
 - Function: No description available'
* hasDocument
 - Returns: boolean
 - Function: Returns true if the document has been added, either as a regular request
or an 'any' request.'
* hasDocument
 - Returns: boolean
 - Function: Returns true if the document has been added, either as a regular request
or an 'any' request.'
* isEmpty
 - Returns: boolean
 - Function: Checks if these requirements are empty.'
* setValidDocCodes
 - Returns: void
 - Function: No description available'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
```

```
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
```

```
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
throwsDocumentIsFaUnacceptableExceptionParam
Purpose: This binding can be set within a Groovy script to control whether a subsequent call to
IReceivedDoc.isAcceptable() or IReceivedDocDocumentAPI.isAcceptable() triggers an exception or not when a
document is unnaceptable. The default value is true.
Type: Boolean
Available Methods:
No methods available
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
```

```
    // Operation [number]: [name of calculation]
    // Formula: [complete mathematical expression]
    // Variables:
    // - [each variable and its purpose]
    // - [data types and units]
    // Steps:
    // 1. [detailed step-by-step breakdown]
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
   ADDITIONAL REQUIREMENTS:
   1. Complex Logic Sections:
    - Must break down multi-step operations
    - Explain interactions between components
    - Detail data flow and transformations
   2. Error Handling:
    - Document all error conditions
    - Explain recovery procedures
    - Detail logging and reporting
   3. API Interactions:
    - Document all platform method calls
    - Explain data transformations
    - Detail error handling
   4. Financial Aid Impact:
    - Explain effects on aid calculations
    - Document regulatory compliance
    - Detail student eligibility impact
   Example Section With All Aspects:
   // CONDITIONALS:
   // Check 1: Payment Period Status Validation
   // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
   // Purpose: Ensure only valid, active payment periods are included
   // Logic:
   // - Retrieve all loan payment periods
   // - Filter based on status (!= "CANCELED")
   // - Compare start dates with reference date
   // Success Path: Period included in calculations
   // Failure Path: Period excluded from totals
   // Business Rules: Canceled periods must not affect grade level
   // Error Handling: Invalid periods logged and skipped
   // Financial Impact: Affects available loan amounts
   //
   // BOUNDARY CHECKS:
   // Check 1: Payment Period Date Validation
   // Source: Academic Calendar Configuration
   // Variables: period.getStartDate(), referenceDate
   // Validation:
   // - Compare period start date with reference date
   // - Ensure dates are not null
   // Edge Cases:
   // - Null dates are excluded
   // - Future dates are filtered
   // Error Handling: Invalid dates logged and skipped
   // Impact: Prevents future periods from affecting current calculation
   //
   // ARITHMETIC:
   // Operation 1: Payment Period Units Summation
   // Formula: sum(validPeriods.map(period => period.getUnits()))
   // Variables:
   // - period.getUnits(): Decimal, represents credit hours
   // - validPeriods: Array of filtered payment periods
   // Steps:
```

```
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
N/A - There is no baseline config. This is a placeholder.
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# ISIR Discrepancy Additional Documents Required

Name: isirDiscrepancyAdditionalDocumentsRequired

Purpose: Determine whether additional documentation should be collected.

In Baseline Config: Yes

Return Type: Void

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "isirDiscrepancyAdditionalDocumentsRequired"
Primary Function: Determine whether additional documentation should be collected for a student in order to
 ascertain whether an ISIR
correction is needed. If additional documentation is needed, this script can also trigger document requests
 for
the aforementioned documentation.
Expected Output Type: Void
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
```

```
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
documentRequest
Purpose: Binding that shall be populated by the script with the selected documents.
Type: IIsirRequiredDocsAPI
Available Methods:
* addAnyDocuments
 - Returns: void
 - Function: Adds a list of documents, any one in the list is required.'
* addAnyDocuments
 - Returns: void
 - Function: Adds a list of documents, any one in the list is required.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* any
 - Returns: IIsirRequiredDocsAPI
 - Function: No description available'
* clear
 - Returns: void
 - Function: Clears all docs.'
* doc
 - Returns: IIsirRequiredDocsAPI
 - Function: No description available'
* getAdditionalDocRequestMessages
 - Returns: List<String>
 - Function: No description available'
* getValidDocCodes
 - Returns: Set<String>
 - Function: No description available'
* hasDocument
 - Returns: boolean
 - Function: Returns true if the document has been added, either as a regular request
or an 'any' request.'
* hasDocument
 - Returns: boolean
 - Function: Returns true if the document has been added, either as a regular request
or an 'any' request.'
* isEmpty
 - Returns: boolean
 - Function: Checks if these requirements are empty.'
* setValidDocCodes
 - Returns: void
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
```

```
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
```

```
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
```

```
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isirRecord
Purpose: The ISIR being evaluated.
Type: IIsirRecordAPI
Available Methods:
* getActiveIsirForAwardYear
 - Returns: Boolean
 - Function: Gets the active ISIR for award year status.'
* getAllFields
 - Returns: @NonNull Map<IsirField<?>,IIsirFieldValue<?>>
 - Function: Retrieves all field values (WARNING: can only be used with 2025+ ISIRs).'
* getAssumptions
 - Returns: List<IAssumptionAPI>
 - Function: Get the assumptions.'
* getAwardYear
 - Returns: String
 - Function: Gets the award year.'
* getCommentCodes
 - Returns: List<ICommentCodeAPI>
 - Function: Get the comment codes.'
* getCpsTransNumber
 - Returns: String
 - Function: Gets the CPS transaction number.'
* getCurrentSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's current SSN.'
* getDiscrepancies
 - Returns: List<IDiscrepancyAPI>
 - Function: Get the discrepancies.'
* getField
 - Returns: <T> @NonNull IIsirFieldValue<T>
 - Function: Retrieves the value of a given field (WARNING: can only be used with 2025+ ISIRs).'
* getFirstName
 - Returns: String
 - Function: Gets the student's first name.'
* getHighlightFlags
 - Returns: List<IHighlightFlagAPI>
 - Function: Get the highlight flags.'
* getIsirFieldValue
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValueOrNull
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValues
 - Returns: Map<String,String>
```

```
 - Function: Get a map containing all ISIR fields (WARNING: This method can only be used up to 2024 award
 year).'
* getIsirIdOriginalNameId
 - Returns: String
 - Function: Gets the student's original name ID.'
* getIsirIdSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's original SSN.'
* getIsirIdYearIndicator
 - Returns: String
 - Function: Gets the ISIR's year indicator.'
* getIsirStatus
 - Returns: String
 - Function: Gets the ISIR status.'
* getIsirStatusBeforeCorrection
 - Returns: String
 - Function: When ISIR Status is Corrected returns the previous ISIR Status.'
* getLastName
 - Returns: String
 - Function: Gets the student's last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the student's middle name.'
* getReceivedDate
 - Returns: Date
 - Function: Gets the system receipt date of the ISIR.'
* getRejectCodes
 - Returns: List<IRejectCodeAPI>
 - Function: Get the reject codes.'
* getSchoolCode1
 - Returns: String
 - Function: Gets the school code 1.'
* getSchoolCode10
 - Returns: String
 - Function: Gets the school code 10.'
* getSchoolCode2
 - Returns: String
 - Function: Gets the school code 2.'
* getSchoolCode3
 - Returns: String
 - Function: Gets the school code 3.'
* getSchoolCode4
 - Returns: String
 - Function: Gets the school code 4.'
* getSchoolCode5
 - Returns: String
 - Function: Gets the school code 5.'
* getSchoolCode6
 - Returns: String
 - Function: Gets the school code 6.'
* getSchoolCode7
 - Returns: String
 - Function: Gets the school code 7.'
* getSchoolCode8
 - Returns: String
 - Function: Gets the school code 8.'
* getSchoolCode9
 - Returns: String
 - Function: Gets the school code 9.'
* isCompatibleWithPackaging
 - Returns: boolean
 - Function: Checks if this ISIR satisfies minimal acceptance criteria for automatic Financial
Aid Package Calculation.'
* isUnverified
 - Returns: boolean
 - Function: Gets the ISIR verification status.'
```

```
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
previousIsirRecords
Purpose: Previous ISIR records with the same award year as the ISIR being evaluated, ordered by descending
 CPS numbers.
Type: List
Available Methods:
No methods available
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
```

```
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Active documents for the student for the award year of the ISIR being evaluated.
Type: IReceivedDocs
Available Methods:
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and Student, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and owners, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code, owners, and subtype, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and Student, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and owners, if it exists.'
* getValidDocCodes
 - Returns: Set<String>
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: Return true if doc exists with the right code and owners'
* noDocs
 - Returns: boolean
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
```

```
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
```

```
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
throwsDocumentIsFaUnacceptableExceptionParam
Purpose: This binding can be set within a Groovy script to control whether a subsequent call to
IReceivedDoc.isAcceptable() or IReceivedDocDocumentAPI.isAcceptable() triggers an exception or not when a
document is unnaceptable. The default value is true.
Type: Boolean
Available Methods:
No methods available
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
allAwardYearReceivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
disbursements
```

```
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
```

```
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
```

```
    // - [step-by-step breakdown of the condition]
    // - [how variables are used]
    // - [any transformations applied]
    // Success Path: [what happens when condition is true]
    // Failure Path: [what happens when condition is false]
    // Business Rules: [specific rules being enforced]
    // Error Handling: [how errors are managed]
    // Financial Impact: [effect on aid calculations]
    // If N/A, explicitly state: "CONDITIONALS: N/A"
  2. BOUNDARY CHECKS Section:
    Format:
    // BOUNDARY CHECKS:
    // Check [number]: [specific limit being enforced]
    // Source: [where boundary values come from]
    // Variables: [variables and methods involved]
    // Validation:
    // - [how boundaries are checked]
    // - [validation method details]
    // Edge Cases:
    // - [specific edge cases handled]
    // - [how each is managed]
    // Error Handling: [boundary violation management]
    // Impact: [effect of boundary violations]
    // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
  3. ARITHMETIC Section:
    Format:
    // ARITHMETIC:
    // Operation [number]: [name of calculation]
    // Formula: [complete mathematical expression]
    // Variables:
    // - [each variable and its purpose]
    // - [data types and units]
    // Steps:
    // 1. [detailed step-by-step breakdown]
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
  ADDITIONAL REQUIREMENTS:
  1. Complex Logic Sections:
    - Must break down multi-step operations
    - Explain interactions between components
    - Detail data flow and transformations
  2. Error Handling:
    - Document all error conditions
    - Explain recovery procedures
    - Detail logging and reporting
  3. API Interactions:
    - Document all platform method calls
    - Explain data transformations
    - Detail error handling
  4. Financial Aid Impact:
    - Explain effects on aid calculations
    - Document regulatory compliance
    - Detail student eligibility impact
  Example Section With All Aspects:
  // CONDITIONALS:
  // Check 1: Payment Period Status Validation
  // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
  // Purpose: Ensure only valid, active payment periods are included
  // Logic:
  // - Retrieve all loan payment periods
  // - Filter based on status (!= "CANCELED")
```

```
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.IsirDiscrepancyAdditionalDocumentsRequire
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
//Variable definitions
def isirSTUDENT_PARENTKILLEDINTHELINEOFDUTY =
 isirRecord.getField(IsirField.STUDENT_PARENTKILLEDINTHELINEOFDUTY).getOrNull()
def isirFPS_IASGINDICATOR = isirRecord.getField(IsirField.FPS_IASGINDICATOR).getOrNull()
def isirFPS_CHILDRENOFFALLENHEROESINDICATOR =
 isirRecord.getField(IsirField.FPS_CHILDRENOFFALLENHEROESINDICATOR).getOrNull()
//If the student has identified parent was killed in line of duty, and the IASG indicator/children of fallen
 heroes indicator are not in an eligible state
if (isirSTUDENT_PARENTKILLEDINTHELINEOFDUTY == "1" && !["1", "2"].contains(isirFPS_IASGINDICATOR) && !["1",
 "2"].contains(isirFPS_CHILDRENOFFALLENHEROESINDICATOR))
{
 //If document exists and is acceptable, then don't request it again
 if (receivedDocuments.hasDoc("IASG-CHF-SelfDeclaration") && receivedDocuments.get("IASG-CHF-
SelfDeclaration","Student").isAcceptable())
 {
 return
 }
 //If document does not exist or is not acceptable, then request the document
 else
 {
 documentRequest.addDocument("IASG-CHF-SelfDeclaration")
 }
}
```
## COMMENT GENERATION REQUIREMENTS:
```

```
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Financial Aid Calculation Scripts

These are the prompts for scripts handling financial aid calculations, including grade levels, award amounts, and loan eligibility.

## Grade Level Progression

Name: gradeLevelProgression

Purpose: Calculate the grade level.

In Baseline Config: Yes

Return Type: Integer

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "gradeLevelProgression"
Primary Function: Calculate the grade level.
Expected Output Type: Integer
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
actualStartDate
```

```
Purpose: The student's actual start date.
Type: LocalDate
Available Methods:
No methods available
acy
Purpose: The academic year of the funding element being evaluated.
Type: IAcademicYearScheduleAPI
Available Methods:
* getCompletedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getPeriodOfEnrollmentEndDate
 - Returns: LocalDate
 - Function: No description available'
* getPeriodOfEnrollmentStartDate
 - Returns: LocalDate
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
acyNo
Purpose: The number of the academic year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
acyOrder
Purpose: The order number of the academic year of the funding element being evaluated, among all non-
canceled academic
years.
Type: Integer
Available Methods:
No methods available
acys
Purpose: Provides access to all non-canceled academic years.
Type: IAcademicYearsScheduleAPI
Available Methods:
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
```

```
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
awy
Purpose: The award year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
awyPeriod
Purpose: The period of the award year of the funding element being evaluated.
Type: IAwardYearAPI
Available Methods:
* asList
 - Returns: IAwardYearsAPI
 - Function: Returns an IAwardYearsAPI instance containing only this Award Year.'
* getEndYear
 - Returns: int
 - Function: Gets the end year of this Award Year, for example 2023 for Award Year 2022-2023.'
* getNext
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that follows this one chronologically.'
* getPrevious
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that precedes this one chronologically.'
* getStartYear
 - Returns: int
 - Function: Gets the start year of this Award Year, for example 2022 for Award Year 2022-2023.'
* getYear
 - Returns: int
 - Function: Short for getEndYear().'
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
No methods available
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
```

```
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
```

```
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
fundCode
Purpose: The fund code of the funding element being evaluated.
Type: String
Available Methods:
No methods available
fundType
```

```
Purpose: The fund type of the funding element being evaluated.
Type: String
Available Methods:
No methods available
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isFirstAcy
Purpose: True if the academic year of the funding element being evaluated is the first one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isLastAcy
Purpose: True if the academic year of the funding element being evaluated is the last one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
```

```
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
lp
Purpose: The loan period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
lpNo
Purpose: The number of the loan period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
pp
Purpose: The payment period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
ppNo
Purpose: The number of the payment period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
```

```
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
```

```
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
```

```
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
scope
```

```
Purpose: Gathers all the scope details of the script being evaluated (acyNo, fundCode,
fundType, awy, lpNo, ppNo) allowing quick access to financialPlan/disbursements/etc
subsets directly pertaining to this scope.
Type: IFundScopeAPI
Available Methods:
* getAcyNo
 - Returns: int
 - Function: No description available'
* getAwy
 - Returns: int
 - Function: No description available'
* getFundCode
 - Returns: String
 - Function: No description available'
* getFundType
 - Returns: String
 - Function: No description available'
* getLpNo
 - Returns: int
 - Function: No description available'
* getPpNo
 - Returns: int
 - Function: No description available'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
```

```
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
Available Methods:
* getPeriodsIntersectingWithGroupForThePeriod
 - Returns: List<? extends IPeriod>
 - Function: Find the periods student was in the given group in the specified period of time'
* isInGroupAtAnyPointInThePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period at any point of time.'
* isInGroupAtDate
 - Returns: boolean
 - Function: Checks if student was in the given group on the given date.'
* isInGroupDuringCompletePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period for the whole time.'
* numberOfDaysInGroupForThePeriod
 - Returns: int
 - Function: Count number of days student was in the given group in the specified period of time'
* getActiveGroupsAtAnyPointInThePeriod
 - Returns: List<String>
 - Function: Find the student's groups active at any point during specified period of time'
* getActiveGroupsAtDate
 - Returns: List<String>
 - Function: Find the student's active groups at the given date'
* getActiveGroupsDuringCompletePeriod
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
term
Purpose: The term of the funding element being evaluated.
```

```
Type: IAcademicTermPeriodAPI
Available Methods:
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getName
 - Returns: String
 - Function: No description available'
* isAttending
 - Returns: boolean
 - Function: No description available'
* isNotAttending
 - Returns: boolean
 - Function: No description available'
* isStandard
 - Returns: boolean
 - Function: No description available'
* isSummer
 - Returns: boolean
 - Function: No description available'
* isSummerHeader
 - Returns: boolean
 - Function: No description available'
* isSummerTrailer
 - Returns: boolean
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
termNo
Purpose: The number of the term of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
```

```
    // - [step-by-step breakdown of the condition]
    // - [how variables are used]
    // - [any transformations applied]
    // Success Path: [what happens when condition is true]
    // Failure Path: [what happens when condition is false]
    // Business Rules: [specific rules being enforced]
    // Error Handling: [how errors are managed]
    // Financial Impact: [effect on aid calculations]
    // If N/A, explicitly state: "CONDITIONALS: N/A"
  2. BOUNDARY CHECKS Section:
    Format:
    // BOUNDARY CHECKS:
    // Check [number]: [specific limit being enforced]
    // Source: [where boundary values come from]
    // Variables: [variables and methods involved]
    // Validation:
    // - [how boundaries are checked]
    // - [validation method details]
    // Edge Cases:
    // - [specific edge cases handled]
    // - [how each is managed]
    // Error Handling: [boundary violation management]
    // Impact: [effect of boundary violations]
    // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
  3. ARITHMETIC Section:
    Format:
    // ARITHMETIC:
    // Operation [number]: [name of calculation]
    // Formula: [complete mathematical expression]
    // Variables:
    // - [each variable and its purpose]
    // - [data types and units]
    // Steps:
    // 1. [detailed step-by-step breakdown]
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
  ADDITIONAL REQUIREMENTS:
  1. Complex Logic Sections:
    - Must break down multi-step operations
    - Explain interactions between components
    - Detail data flow and transformations
  2. Error Handling:
    - Document all error conditions
    - Explain recovery procedures
    - Detail logging and reporting
  3. API Interactions:
    - Document all platform method calls
    - Explain data transformations
    - Detail error handling
  4. Financial Aid Impact:
    - Explain effects on aid calculations
    - Document regulatory compliance
    - Detail student eligibility impact
  Example Section With All Aspects:
  // CONDITIONALS:
  // Check 1: Payment Period Status Validation
  // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
  // Purpose: Ensure only valid, active payment periods are included
  // Logic:
  // - Retrieve all loan payment periods
  // - Filter based on status (!= "CANCELED")
```

```
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.GradeLevelProgressionScript)
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
def referenceDate = program.isTerm() ? term.getStartDate() : acy.getStartDate();
def totalPrevAcyUnits = acys
 .findAll { it.getNumber() < acyNo }
 .collect { it.getStatus() == "ACTUAL" ? it.getCompletedUnits() : it.getUnits() }
 .sum(0.0);
def totalCurAcyUnits = acy.getLoanPaymentPeriods()
 .findAll { it.getStatus() != "CANCELED" && it.getStartDate() < referenceDate }
 .collect { it.getUnits() }
 .sum(0.0);
def totalCreditsCompletedBeforeProgramStartDate = program.getCourses()
 .findAll {
 it.getSchedulingStatus() == "PASSED" &&
 it.getEndDate() < actualStartDate &&
 (it.getIncompleteResolutionDate() == null || it.getIncompleteResolutionDate() < actualStartDate)
 }
 .collect { it.getUnits() }
 .sum(0.0);
def totalCreditUnits = (program.getTotalAcceptedTransferUnits() +
 program.getAssessedUnits() +
 totalPrevAcyUnits +
 totalCurAcyUnits +
 totalCreditsCompletedBeforeProgramStartDate);
def creditPerAydUnit = totalCreditUnits / program.getAydUnits();
def baseGradeLevel = (program.isGraduate()
 ? 6
```
```

```
 : program.getProgramType() == "CERTIFICATE_POST_BACCALAUREATE"
 ? 5
 : 1);
def unboundedGradeLevel = baseGradeLevel + creditPerAydUnit.intValue();
def isAssociate = program.getProgramType() == "ASSOCIATE";
def gradeLevelUb = (isAssociate
 ? 2
 : program.isUndergraduate()
 ? 5
 : 7);
def gradeLevel = [unboundedGradeLevel, gradeLevelUb].min();
log.debug("actualStartDate={}, referenceDate={}, totalPrevAcyUnits={}, totalCurAcyUnits={},
 totalCreditsCompletedBeforeProgramStartDate={}, totalCreditUnits={}, creditPerAydUnit={},
 baseGradeLevel={}, isAssociate={}, gradeLevelUb={}, gradeLevel={}",
 actualStartDate,
 referenceDate,
 totalPrevAcyUnits,
 totalCurAcyUnits,
 totalCreditsCompletedBeforeProgramStartDate,
 totalCreditUnits,
 creditPerAydUnit,
 baseGradeLevel,
 isAssociate,
 gradeLevelUb,
 gradeLevel
);
return gradeLevel;
```
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Award Overriding Criteria

Name: awardOverridingCriteria

Purpose: Override award amounts.

In Baseline Config: Yes

Return Type: IAwardOverrideAPI

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "awardOverridingCriteria"
```

```
Primary Function: Override award amounts.
Expected Output Type: IAwardOverrideAPI
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
acy
Purpose: The academic year of the funding element being evaluated.
Type: IAcademicYearScheduleAPI
Available Methods:
* getCompletedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getPeriodOfEnrollmentEndDate
 - Returns: LocalDate
 - Function: No description available'
* getPeriodOfEnrollmentStartDate
 - Returns: LocalDate
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
acyNo
Purpose: The number of the academic year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
acyOrder
Purpose: The order number of the academic year of the funding element being evaluated, among all non-
canceled academic
years.
Type: Integer
Available Methods:
```

```
No methods available
acys
Purpose: Provides access to all non-canceled academic years.
Type: IAcademicYearsScheduleAPI
Available Methods:
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
awardInfo
Purpose: Binding used to override award amounts. This binding must be updated using one of its methods and
 then
returned by the script.
Type: IAwardOverrideAPI
Available Methods:
* getComment
 - Returns: String
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getRetainedAmount
 - Returns: BigDecimal
 - Function: No description available'
* withComment
 - Returns: IAwardOverrideAPI
 - Function: No description available'
* withMaxAmount
 - Returns: IAwardOverrideAPI
 - Function: No description available'
* withRetainedAmount
 - Returns: IAwardOverrideAPI
 - Function: No description available'
awy
Purpose: The award year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
awyPeriod
Purpose: The period of the award year of the funding element being evaluated.
Type: IAwardYearAPI
Available Methods:
* asList
 - Returns: IAwardYearsAPI
 - Function: Returns an IAwardYearsAPI instance containing only this Award Year.'
* getEndYear
 - Returns: int
 - Function: Gets the end year of this Award Year, for example 2023 for Award Year 2022-2023.'
* getNext
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that follows this one chronologically.'
* getPrevious
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that precedes this one chronologically.'
* getStartYear
 - Returns: int
```

```
 - Function: Gets the start year of this Award Year, for example 2022 for Award Year 2022-2023.'
* getYear
 - Returns: int
 - Function: Short for getEndYear().'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
No methods available
disbursements
```

```
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
```

```
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollmentStatus
Purpose: The student's enrollment status. Can be one of FULL_TIME, THREE_QUARTER_TIME, HALF_TIME,
LESS_THAN_HALF_TIME or NOT_ATTENDING.
Type: String
Available Methods:
No methods available
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
```

```
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
```

```
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCode
Purpose: The fund code of the funding element being evaluated.
Type: String
Available Methods:
No methods available
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
fundType
Purpose: The fund type of the funding element being evaluated.
Type: String
Available Methods:
No methods available
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
```

```
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isFirstAcy
Purpose: True if the academic year of the funding element being evaluated is the first one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isLastAcy
Purpose: True if the academic year of the funding element being evaluated is the last one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
```

```
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
lp
Purpose: The loan period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
lpNo
Purpose: The number of the loan period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
optimizer
Purpose: Binding that allows scripts to affect the optimization of DSUB, DUNSUB,
PLUS, and GPLUS funds. IMPORTANT: this bind is optional, which means
that scripts that process any other funds will not have this binding
declared at all.
Type: IOptimizerConfigAPI
Available Methods:
* addPhase
 - Returns: IOptimizerConfigAPI
 - Function: Same as addPhase(int) but only actually does anything if a given
condition is true.'
* addPhase
 - Returns: IOptimizerConfigAPI
 - Function: Same as addPhase(int, UnaryOperator) but only actually does anything
if a given condition is true.'
* addPhase
 - Returns: IOptimizerConfigAPI
 - Function: Same as addPhase(int, UnaryOperator) but using only IOptimizationPhaseAPI defaults.'
* addPhase
 - Returns: IOptimizerConfigAPI
 - Function: Adds an Optimization Phase for the Paymemt Period/Term being evaluated.'
* getFutureVariables
 - Returns: List<IScheduledPeriodIdentityAPI>
 - Function: Gets the identities of Payment Periods/Terms in future academic years (in
relation to the one being calculated) that are implied in this Fund Optimization.'
* setCompensationValue
 - Returns: IOptimizerConfigAPI
 - Function: Sets a Compensation Value for this Payment Period/Term (PP for simplicity here).'
* setCumulativeGradeLevelLimitApplicable
 - Returns: IOptimizerConfigAPI
 - Function: Declares whether this Payment Period/Term should be included in the
Cumulative Grade Level Limit Constraint.'
* setFundingForPeriodFilter
 - Returns: IOptimizerConfigAPI
 - Function: Allows selecting which higher-priority funds will influence the Term
Cost of Attendance (COA) constraint.'
partialFundOutput
Purpose: The current version of the funding information for the Academic Year + fund being calculated.
This will eventually be incorporated to the financialPlan binding.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
```

```
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
period
Purpose: A period matching the payment period being evaluated.
Type: IPeriod
Available Methods:
No methods available
pp
Purpose: The payment period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
```

```
ppNo
Purpose: The number of the payment period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
```

```
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
```

```
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
```

```
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
sapStatus
Purpose: The student's SAP status. Can be one of RG_REGULAR, FW_FINANCIAL_AID_WARNING,
FD_FINANCIAL_AID_DISQUALIFICATION or FP_FINANCIAL_AID_PROBATION.
Type: String
Available Methods:
No methods available
scope
Purpose: Gathers all the scope details of the script being evaluated (acyNo, fundCode,
fundType, awy, lpNo, ppNo) allowing quick access to financialPlan/disbursements/etc
subsets directly pertaining to this scope.
Type: IFundScopeAPI
Available Methods:
* getAcyNo
 - Returns: int
 - Function: No description available'
* getAwy
 - Returns: int
 - Function: No description available'
* getFundCode
 - Returns: String
 - Function: No description available'
* getFundType
 - Returns: String
 - Function: No description available'
* getLpNo
 - Returns: int
 - Function: No description available'
* getPpNo
 - Returns: int
 - Function: No description available'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
```

```
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
```

```
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
Available Methods:
* getPeriodsIntersectingWithGroupForThePeriod
 - Returns: List<? extends IPeriod>
 - Function: Find the periods student was in the given group in the specified period of time'
* isInGroupAtAnyPointInThePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period at any point of time.'
* isInGroupAtDate
 - Returns: boolean
 - Function: Checks if student was in the given group on the given date.'
* isInGroupDuringCompletePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period for the whole time.'
* numberOfDaysInGroupForThePeriod
 - Returns: int
 - Function: Count number of days student was in the given group in the specified period of time'
* getActiveGroupsAtAnyPointInThePeriod
 - Returns: List<String>
 - Function: Find the student's groups active at any point during specified period of time'
* getActiveGroupsAtDate
 - Returns: List<String>
 - Function: Find the student's active groups at the given date'
* getActiveGroupsDuringCompletePeriod
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
term
Purpose: The term of the funding element being evaluated.
Type: IAcademicTermPeriodAPI
Available Methods:
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getName
 - Returns: String
 - Function: No description available'
* isAttending
 - Returns: boolean
 - Function: No description available'
* isNotAttending
 - Returns: boolean
 - Function: No description available'
* isStandard
 - Returns: boolean
 - Function: No description available'
* isSummer
 - Returns: boolean
 - Function: No description available'
* isSummerHeader
 - Returns: boolean
 - Function: No description available'
* isSummerTrailer
 - Returns: boolean
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
termNo
Purpose: The number of the term of the funding element being evaluated.
Type: Integer
Available Methods:
```

```
No methods available
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
 // Operation [number]: [name of calculation]
 // Formula: [complete mathematical expression]
 // Variables:
 // - [each variable and its purpose]
 // - [data types and units]
 // Steps:
 // 1. [detailed step-by-step breakdown]
 // 2. [intermediate calculations]
 // 3. [transformations applied]
```

```
 // Business Context: [why this calculation is needed]
 // Result: [meaning of the output]
 // Financial Impact: [effect on aid calculations]
 // Limitations: [any mathematical constraints]
 // If N/A, explicitly state: "ARITHMETIC: N/A"
ADDITIONAL REQUIREMENTS:
1. Complex Logic Sections:
 - Must break down multi-step operations
 - Explain interactions between components
 - Detail data flow and transformations
2. Error Handling:
 - Document all error conditions
 - Explain recovery procedures
 - Detail logging and reporting
3. API Interactions:
 - Document all platform method calls
 - Explain data transformations
 - Detail error handling
4. Financial Aid Impact:
 - Explain effects on aid calculations
 - Document regulatory compliance
 - Detail student eligibility impact
Example Section With All Aspects:
// CONDITIONALS:
// Check 1: Payment Period Status Validation
// Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
```
```

```
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.AwardOverridingCriteriaScript)
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
import org.joda.time.LocalDate
/* Baseline - Award Overriding Criteria for DSUB DUNSUB - combined Term and Non-Term */
static <T> Closure<T> lazy(Closure<T> init) {
 T value
 boolean isInitDue = true
 return {
 if (isInitDue) {
 isInitDue = false
 value = init()
 }
 value
 }
}
def disbursementsInThisPp = lazy {
 disbursements.getInScope(scope)
}
def nonCancelledDisbursementsInThisPp = lazy {
 disbursementsInThisPp()
 .getCancelled(false)
 .filterStatuses { status -> status != "DISBURSEMENT_CANCELED" }
}
def disbursedDisbursementsInThisPp = lazy {
 nonCancelledDisbursementsInThisPp().getWithStatus("DISBURSED")
}
def thisPpIsDisbursed = lazy {
 log.debug(!disbursedDisbursementsInThisPp().empty, "DISBURSED PP")
}
def thisPpFundOutput = lazy {
 partialFundOutput.getInLoanPeriod(lpNo).getInPaymentPeriod(ppNo)
}
def thisPpIsProrated = lazy {
 log.debug(thisPpFundOutput().eligible && thisPpFundOutput().prorated, "PRORATED PP")
}
def isInNASuffix = term && term.standard && acy.terms.findAll { t -> t.startDate >= term.startDate &&
 t.standard }
 .every { t -> t.notAttending }
def disbursedAmount = thisPpIsDisbursed()
 ? disbursedDisbursementsInThisPp().totalDisbursementAmount
 : (BigDecimal) null
def acceptedAmount = fundAcceptance.getInScope(scope)
 .filterStatuses {it != "PENDING_ACCEPTANCE" }
 .let {it.empty ? null : it.validatedAcceptedAmount }
def effectiveAcceptedAmount = thisPpIsDisbursed() ? disbursedAmount : acceptedAmount
def now = LocalDate.now()
def activeSummerTerms = acy.terms.filter {
 it.summer && program.courses.getAssociatedTo(it)
 .any {
 it.schedulingStatus != "PROJECTED"
 }
}
def activeSummerTrailers = activeSummerTerms.filter {
 it.summerTrailer
}
def summerTrailerHasCourses = !activeSummerTrailers.empty
def summerTrailerStart = summerTrailerHasCourses ? activeSummerTrailers.startDate : null
def isActiveSummerTerm = activeSummerTerms.contains(term)
def isBeyondSummerTrailerThreshold = summerTrailerStart && now >= summerTrailerStart.minusDays(60)
def summerDocExists = lazy {
 def summerDocsInAwy = receivedDocuments.forAwardYear(awy)
 .getDocumentsForDocCode("SummerApplication", "Student")
 log.debug(summerDocsInAwy != null && !summerDocsInAwy.empty, "SUMMER DOC EXISTS")
```

```
}
def isSummerTrailerState = log.debug(summerTrailerHasCourses && isBeyondSummerTrailerThreshold &&
 summerDocExists(), "SUMMER TRAILER STATE")
def isActiveSummerHeader = log.debug(term?.summerHeader && isActiveSummerTerm, "ACTIVE SUMMER HEADER")
def isLessThanHalfTime = enrollmentStatus == "LESS_THAN_HALF_TIME"
def standardOrSummerHeader = (term?.isStandard() || isActiveSummerHeader)
optimizer
 .setCumulativeGradeLevelLimitApplicable(
 (!term || standardOrSummerHeader)
 && (!isLastAcy || program.undergraduate)
 && (!isLastAcy || !isInNASuffix)
 && !thisPpIsProrated())
 .setFundingForPeriodFilter { fundCode, fundType -> fundCode != "DISCOUNT" }
 .setCompensationValue(0)
 .addPhase(thisPpIsDisbursed(), 1) {
 it.withGiven(disbursedAmount)
 }
if (!isLessThanHalfTime) {
 optimizer
 .addPhase(!term, 2)
 .addPhase(term && standardOrSummerHeader && (!isSummerTrailerState || effectiveAcceptedAmount), 2) {
 it.withPpMaximum(isSummerTrailerState ? effectiveAcceptedAmount : null)
 }
 .addPhase(isSummerTrailerState && (isActiveSummerTerm || standardOrSummerHeader && effectiveAcceptedAmount
 == null), 3)
}
if (program.enrollmentStatus == "X" || pp.status == "CANCELED")
{
 return awardInfo.withMaxAmount(0.0).withRetainedAmount(0.0)
}
if (program.term)
{
 def overlappingTerms = primaryProgram.terms.getOverlappingWith(term)
 /* Methods returning collection, as well as filters, will be plural; otherwise singular */
 if (overlappingTerms.studentsTermStatuses.contains("WITHDRAWN"))
 {
 log.debug("STUDENT TERM IS WITHDRAWN")
 def r2t4OverlappingTerms = r2t4.getOverlappingWith(term)
 .getWithProcessStatuses(["NOT_REQUIRED", "COMPLETED"])
 .getWithOldas(overlappingTerms.oldas)
 if (!r2t4OverlappingTerms.empty)
 {
 log.debug("R2T4 Overlapping Terms is not empty, FREEZING!")
 def sum = nonCancelledDisbursementsInThisPp().ppMaxDisbursementAmount
 return awardInfo.withMaxAmount(sum).withRetainedAmount(sum)
 }
 else
 {
 log.debug("R2T4 PROCESS NOT YET IN A FREEZING STATE")
 }
 }
 if (enrollmentStatus == "NOT_ATTENDING")
 {
 log.debug("ZEROING - enrollmentStatus={} ", enrollmentStatus)
 return awardInfo.withMaxAmount(0.0).withRetainedAmount(0.0)
 }
}
if (log.debug(lp.endDate <= now, "LOAN PERIOD IN THE PAST")
 || (thisPpIsProrated() && thisPpIsDisbursed()))
{
 def frozenAmount = disbursedAmount == null ? 0.0 : disbursedAmount
 log.debug("FREEZING TO {}", frozenAmount)
 return awardInfo.withMaxAmount(frozenAmount).withRetainedAmount(frozenAmount)
}
return awardInfo
```

```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Award Year Selection Criteria

Name: awardYearSelectionCriteria

Purpose: Determine the award year to use in crossover payment periods.

In Baseline Config: Yes

Return Type: IScriptedCrossoverArbitratorResult

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "awardYearSelectionCriteria"
Primary Function: Determine the award year to use in crossover payment periods.
Expected Output Type: IScriptedCrossoverArbitratorResult
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
acy
Purpose: The academic year of the funding element being evaluated.
Type: IAcademicYearScheduleAPI
Available Methods:
* getCompletedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
```

```
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getPeriodOfEnrollmentEndDate
 - Returns: LocalDate
 - Function: No description available'
* getPeriodOfEnrollmentStartDate
 - Returns: LocalDate
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
acyNo
Purpose: The number of the academic year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
acyOrder
Purpose: The order number of the academic year of the funding element being evaluated, among all non-
canceled academic
years.
Type: Integer
Available Methods:
No methods available
acys
Purpose: Provides access to all non-canceled academic years.
Type: IAcademicYearsScheduleAPI
Available Methods:
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
awy
Purpose: The award year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
awyPeriod
Purpose: The period of the award year of the funding element being evaluated.
Type: IAwardYearAPI
Available Methods:
```

```
* asList
 - Returns: IAwardYearsAPI
 - Function: Returns an IAwardYearsAPI instance containing only this Award Year.'
* getEndYear
 - Returns: int
 - Function: Gets the end year of this Award Year, for example 2023 for Award Year 2022-2023.'
* getNext
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that follows this one chronologically.'
* getPrevious
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that precedes this one chronologically.'
* getStartYear
 - Returns: int
 - Function: Gets the start year of this Award Year, for example 2022 for Award Year 2022-2023.'
* getYear
 - Returns: int
 - Function: Short for getEndYear().'
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
No methods available
crossoverHelper
Purpose: Binding containing utilities for crossover payment period arbitration.
Type: ICrossoverArbitratorHelper
Available Methods:
* getCurrentAwardYear
 - Returns: int
 - Function: Gets the award year corresponding to the current alternative.'
* getUpcomingAwardYear
 - Returns: int
 - Function: Gets the award year corresponding to the upcoming alternative.'
* hasDisbursedDisbursementsInCurrentAwardYear
 - Returns: boolean
 - Function: Indicates if the student has disbursed disbursements associated to the current award year for
 the academic year
and payment period currently evaluated.'
* hasDisbursedDisbursementsInUpcomingAwardYear
 - Returns: boolean
```

```
 - Function: Indicates if the student has disbursed disbursements associated to the upcoming award year for
 the academic year
and payment period currently evaluated.'
* hasPaymentPeriodLockedInCurrentAwardYear
 - Returns: boolean
 - Function: Indicates if the student has locked disbursements associated to the current award year for the
 academic year and
payment period currently evaluated.'
* hasPaymentPeriodLockedInUpcomingAwardYear
 - Returns: boolean
 - Function: Indicates if the student has locked disbursements associated to the upcoming award year for the
 academic year and
payment period currently evaluated.'
* isReEntryWithin180Days
 - Returns: boolean
 - Function: Indicates if the student has reentered the program within 180 days and if the student reentered
 within the
academic year currently evaluated.'
* useCurrentAlternative
 - Returns: IScriptedCrossoverArbitratorResult
 - Function: Used in conjunction with a return statement to let the script know it should use the CURRENT
 alternative.'
* useDefaultAlternative
 - Returns: IScriptedCrossoverArbitratorResult
 - Function: Used in conjunction with a return statement to let the script know it should use the DEFAULT
 alternative.'
* useUpcomingAlternative
 - Returns: IScriptedCrossoverArbitratorResult
 - Function: Used in conjunction with a return statement to let the script know it should use the UPCOMING
 alternative.'
currentFunds
Purpose: Current funds.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
```

```
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
```

```
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
```

```
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
fundCode
Purpose: The fund code of the funding element being evaluated.
Type: String
Available Methods:
No methods available
fundType
Purpose: The fund type of the funding element being evaluated.
Type: String
Available Methods:
No methods available
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
```

```
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isFirstAcy
Purpose: True if the academic year of the funding element being evaluated is the first one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isLastAcy
Purpose: True if the academic year of the funding element being evaluated is the last one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
lp
Purpose: The loan period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
lpNo
Purpose: The number of the loan period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
pp
Purpose: The payment period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
```

```
Available Methods:
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
ppNo
Purpose: The number of the payment period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
```

```
    - Returns: String
    - Function: No description available'
  * getEnrollmentStatus
    - Returns: String
    - Function: No description available'
  * getEnrollmentStatusEffectiveDate
    - Returns: LocalDate
    - Function: No description available'
  * getEnrollmentStatusSubType
    - Returns: String
    - Function: No description available'
  * getExternalProgramId
    - Returns: String
    - Function: No description available'
  * getFaEligibleIndicator
    - Returns: Boolean
    - Function: No description available'
  * getGpa
    - Returns: BigDecimal
    - Function: No description available'
  * getManualSapEvaluationIndicator
    - Returns: Boolean
    - Function: No description available'
  * getModality
    - Returns: String
    - Function: No description available'
  * getOfficialLastDateOfAttendance
    - Returns: LocalDate
    - Function: No description available'
  * getOpeId
    - Returns: String
    - Function: No description available'
  * getPrimaryLocation
    - Returns: String
    - Function: No description available'
  * getProgramLengthMeasurementUnit
    - Returns: String
    - Function: No description available'
  * getProgramLengthNumberOfUnits
    - Returns: int
    - Function: No description available'
  * getProgramMeasurementUnit
    - Returns: String
    - Function: No description available'
  * getProgramType
    - Returns: String
    - Function: No description available'
  * getRequiredGpa
    - Returns: BigDecimal
    - Function: No description available'
  * getSpecialPrograms
    - Returns: String
    - Function: No description available'
  * getStartDate
    - Returns: LocalDate
    - Function: No description available'
  * getStudentResources
    - Returns: List<IStudentResourceAPI>
    - Function: No description available'
  * getTermAcys
    - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
    - Function: No description available'
  * getTerms
    - Returns: ITermPeriodsAPI
    - Function: No description available'
  * getTermType
```

```
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
```

```
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
scope
Purpose: Gathers all the scope details of the script being evaluated (acyNo, fundCode,
fundType, awy, lpNo, ppNo) allowing quick access to financialPlan/disbursements/etc
subsets directly pertaining to this scope.
Type: IFundScopeAPI
Available Methods:
* getAcyNo
 - Returns: int
 - Function: No description available'
* getAwy
 - Returns: int
 - Function: No description available'
* getFundCode
 - Returns: String
 - Function: No description available'
* getFundType
 - Returns: String
 - Function: No description available'
* getLpNo
 - Returns: int
 - Function: No description available'
* getPpNo
 - Returns: int
 - Function: No description available'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
```

```
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
```

```
Type: IStudentGroupsAPI
Available Methods:
* getPeriodsIntersectingWithGroupForThePeriod
 - Returns: List<? extends IPeriod>
 - Function: Find the periods student was in the given group in the specified period of time'
* isInGroupAtAnyPointInThePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period at any point of time.'
* isInGroupAtDate
 - Returns: boolean
 - Function: Checks if student was in the given group on the given date.'
* isInGroupDuringCompletePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period for the whole time.'
* numberOfDaysInGroupForThePeriod
 - Returns: int
 - Function: Count number of days student was in the given group in the specified period of time'
* getActiveGroupsAtAnyPointInThePeriod
 - Returns: List<String>
 - Function: Find the student's groups active at any point during specified period of time'
* getActiveGroupsAtDate
 - Returns: List<String>
 - Function: Find the student's active groups at the given date'
* getActiveGroupsDuringCompletePeriod
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
term
Purpose: The term of the funding element being evaluated.
Type: IAcademicTermPeriodAPI
Available Methods:
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getName
 - Returns: String
 - Function: No description available'
* isAttending
 - Returns: boolean
 - Function: No description available'
* isNotAttending
 - Returns: boolean
 - Function: No description available'
* isStandard
 - Returns: boolean
 - Function: No description available'
* isSummer
 - Returns: boolean
 - Function: No description available'
* isSummerHeader
 - Returns: boolean
 - Function: No description available'
* isSummerTrailer
 - Returns: boolean
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
termNo
Purpose: The number of the term of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
upcomingFunds
```

```
Purpose: Upcoming funds.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
```

```
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
 // Operation [number]: [name of calculation]
 // Formula: [complete mathematical expression]
 // Variables:
 // - [each variable and its purpose]
 // - [data types and units]
 // Steps:
 // 1. [detailed step-by-step breakdown]
 // 2. [intermediate calculations]
 // 3. [transformations applied]
 // Business Context: [why this calculation is needed]
 // Result: [meaning of the output]
 // Financial Impact: [effect on aid calculations]
 // Limitations: [any mathematical constraints]
 // If N/A, explicitly state: "ARITHMETIC: N/A"
ADDITIONAL REQUIREMENTS:
1. Complex Logic Sections:
 - Must break down multi-step operations
```

```
   - Explain interactions between components
   - Detail data flow and transformations
  2. Error Handling:
   - Document all error conditions
   - Explain recovery procedures
   - Detail logging and reporting
  3. API Interactions:
   - Document all platform method calls
   - Explain data transformations
   - Detail error handling
  4. Financial Aid Impact:
   - Explain effects on aid calculations
   - Document regulatory compliance
   - Detail student eligibility impact
  Example Section With All Aspects:
  // CONDITIONALS:
  // Check 1: Payment Period Status Validation
  // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
  // Purpose: Ensure only valid, active payment periods are included
  // Logic:
  // - Retrieve all loan payment periods
  // - Filter based on status (!= "CANCELED")
  // - Compare start dates with reference date
  // Success Path: Period included in calculations
  // Failure Path: Period excluded from totals
  // Business Rules: Canceled periods must not affect grade level
  // Error Handling: Invalid periods logged and skipped
  // Financial Impact: Affects available loan amounts
  //
  // BOUNDARY CHECKS:
  // Check 1: Payment Period Date Validation
  // Source: Academic Calendar Configuration
  // Variables: period.getStartDate(), referenceDate
  // Validation:
  // - Compare period start date with reference date
  // - Ensure dates are not null
  // Edge Cases:
  // - Null dates are excluded
  // - Future dates are filtered
  // Error Handling: Invalid dates logged and skipped
  // Impact: Prevents future periods from affecting current calculation
  //
  // ARITHMETIC:
  // Operation 1: Payment Period Units Summation
  // Formula: sum(validPeriods.map(period => period.getUnits()))
  // Variables:
  // - period.getUnits(): Decimal, represents credit hours
  // - validPeriods: Array of filtered payment periods
  // Steps:
  // 1. Filter valid payment periods
  // 2. Extract units from each period
  // 3. Sum all units using reduce
  // Business Context: Calculate total completed units for grade level
  // Result: Total units affecting grade level
  // Financial Impact: Determines loan eligibility amounts
  // Limitations: Cannot exceed program maximum units
  ## Groovy Script for Analysis
  ```groovy
  //file:noinspection UnnecessaryQualifiedReference
  @groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.AwardYearSelectionCriteriaScript)
  package CHANGE_ME // this should be updated to your actual package name
  import com.oracle.sfp.scripting.api.*
  import com.oracle.sfp.scripting.api.util.*
  /* Pell Award Year Selection Criteria - combined Term and Non-Term*/
  //variables for earlier and later year ISIRs
```
```

```
def earlierYearISIRExists = isirRecords.getCurrentIsirForAwardYear(crossoverHelper.getCurrentAwardYear()) !=
 null
def laterYearISIRExists = isirRecords.getCurrentIsirForAwardYear(crossoverHelper.getUpcomingAwardYear()) !=
 null
//variables for earlier and later year Pell disbursements that are in "Disbursed" status
def earlierYearPellDisbursementExists = crossoverHelper.hasDisbursedDisbursementsInCurrentAwardYear()
def laterYearPellDisbursementExists = crossoverHelper.hasDisbursedDisbursementsInUpcomingAwardYear()
//if an ISIR and disbursements exist in the earlier year, then select the earlier year
if (earlierYearISIRExists && earlierYearPellDisbursementExists) {
 return crossoverHelper.useCurrentAlternative()
}
//if an ISIR and disbursements exist the later year, then select the later year
if (laterYearISIRExists && laterYearPellDisbursementExists) {
 return crossoverHelper.useUpcomingAlternative()
}
//if an ISIR exists in the earlier year and remaining eligibility > $0 for pell in the earlier year, then
 select the earlier year
if (earlierYearISIRExists && currentFunds.getFund("PELL").getOverlappingWith(pp).maxAmount > 0) {
 return crossoverHelper.useCurrentAlternative()
}
//if an ISIR exists in the later year and no ISIRs exist in the earlier year, then select the later year
if (laterYearISIRExists && !earlierYearISIRExists) {
 return crossoverHelper.useUpcomingAlternative()
}
//if none of the above criteria are met, then use default pell crossover logic
return crossoverHelper.useDefaultAlternative()
```
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# PLUS Credit Decision Matching Criteria

Name: plusCreditDecisionMatchingCriteria

Purpose: Determine the (G)PLUS loans applicable to the term/payment period.

In Baseline Config: Yes

Return Type: List<IPlusLoanAPI>

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "plusCreditDecisionMatchingCriteria"
Primary Function: Determine (G)PLUS loans applicable to the term/payment period being evaluated. If
```

```
this script is not defined the system will default to the hardwired, single-borrower
PLUS calculation logic that predates the inclusion of this feature.
Expected Output Type: List<IPlusLoanAPI>
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
acys
Purpose: Provides access to all non-canceled academic years.
Type: IAcademicYearsScheduleAPI
Available Methods:
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
awy
Purpose: The award year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
awyPeriod
Purpose: The period of the award year of the funding element being evaluated.
Type: IAwardYearAPI
Available Methods:
* asList
 - Returns: IAwardYearsAPI
 - Function: Returns an IAwardYearsAPI instance containing only this Award Year.'
* getEndYear
 - Returns: int
 - Function: Gets the end year of this Award Year, for example 2023 for Award Year 2022-2023.'
* getNext
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that follows this one chronologically.'
* getPrevious
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that precedes this one chronologically.'
* getStartYear
 - Returns: int
 - Function: Gets the start year of this Award Year, for example 2022 for Award Year 2022-2023.'
* getYear
 - Returns: int
 - Function: Short for getEndYear().'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
```

```
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
No methods available
deniedPlusLoans
Purpose: Same as the plusLoans binding except that the loans in this case have had their latest CRSP
denied and have not received yet an accepted CRCS.
Type: IPlusLoansAPI
Available Methods:
No methods available
disbursements
```

```
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
```

```
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
```

```
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
```

```
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCode
Purpose: The fund code of the funding element being evaluated.
Type: String
Available Methods:
No methods available
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
fundType
Purpose: The fund type of the funding element being evaluated.
Type: String
Available Methods:
No methods available
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
```

```
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isFirstAcy
Purpose: True if the academic year of the funding element being evaluated is the first one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isLastAcy
Purpose: True if the academic year of the funding element being evaluated is the last one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
```

```
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
lp
Purpose: The loan period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
lpNo
Purpose: The number of the loan period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
plusLoans
Purpose: Accepted PLUS loans for the student overlapping the loan period being evaluated
but having not been used up in previous academic years. All loans are of one
consistent type (either Graduate or Parent PLUS) according to the calculation
in progress. By default the elements will contain a priority queue consisting
of (i) disbursed status (reversed, which prioritizes disbursed loans and avoids
returns when the summation of loans goes beyond COA), (ii) relative priority in
latest package, (iii) latest known process date (older loans come first) and (iv)
loan amount (reversed, which prioritizes maximum/higher amounts and increases
their likelihood of being fully used but may cause lower-amount loans not to be
packaged due to COA).
Type: IPlusLoansAPI
Available Methods:
No methods available
plusOverrides
Purpose: Contains the PLUS manual overrides for this academic year, only of the relevant
type (either Graduate or Parent PLUS) to the calculation in progress.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
```

```
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
pp
Purpose: The payment period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
ppNo
Purpose: The number of the payment period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
```

```
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
```

```
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
projectedPlusLoan
Purpose: A placeholder for eventual actual PLUS loans, this binding can be added as an
element to the script result to include a Projected (G)PLUS fund in the package
and thus show that the student is generally eligible for PLUS in this program
but still needs to apply for actual loans. The script can also set limits
and priorities to this object, the defaults being the maximum amount up to the
Cost of Attendance in this academic year and the lowest possible priority (i.e.
an empty priority queue).
Type: IProjectedPlusLoanAPI
Available Methods:
No methods available
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
```

```
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
```

```
 - Function: Gets the status for a given period for the student.'
scope
Purpose: Gathers all the scope details of the script being evaluated (acyNo, fundCode,
fundType, awy, lpNo, ppNo) allowing quick access to financialPlan/disbursements/etc
subsets directly pertaining to this scope.
Type: IFundScopeAPI
Available Methods:
* getAcyNo
 - Returns: int
 - Function: No description available'
* getAwy
 - Returns: int
 - Function: No description available'
* getFundCode
 - Returns: String
 - Function: No description available'
* getFundType
 - Returns: String
 - Function: No description available'
* getLpNo
 - Returns: int
 - Function: No description available'
* getPpNo
 - Returns: int
 - Function: No description available'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
```

```
    - Returns: String
    - Function: Gets the first name.'
  * getGender
    - Returns: String
    - Function: Gets the gender.'
  * getHighSchoolEquivalency
    - Returns: Boolean
    - Function: Gets the high school equivalency status.'
  * getLastName
    - Returns: String
    - Function: Gets the last name.'
  * getMiddleName
    - Returns: String
    - Function: Gets the middle name.'
  * getPersonId
    - Returns: Long
    - Function: Gets the person ID.'
  * getProgramType
    - Returns: String
    - Function: Gets the program type.'
  * getRecentFinancialAidHistoryRecords
    - Returns: IRecentFahRecordsAPI
    - Function: Gets the recent financial aid records.'
  * getSchoolCode
    - Returns: String
    - Function: Gets the school code.'
  * getSchoolEmployee
    - Returns: Boolean
    - Function: Gets the school employee status.'
  * getSchoolEmployeeDependent
    - Returns: Boolean
    - Function: Gets the school employee dependent status.'
  * getSsn
    - Returns: ISocialSecurityNumber
    - Function: Gets the SSN.'
  * getStudentId
    - Returns: Long
    - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
Available Methods:
  * getPeriodsIntersectingWithGroupForThePeriod
    - Returns: List<? extends IPeriod>
    - Function: Find the periods student was in the given group in the specified period of time'
  * isInGroupAtAnyPointInThePeriod
    - Returns: boolean
    - Function: Checks if student was in the given group in the specified period at any point of time.'
  * isInGroupAtDate
    - Returns: boolean
    - Function: Checks if student was in the given group on the given date.'
  * isInGroupDuringCompletePeriod
    - Returns: boolean
    - Function: Checks if student was in the given group in the specified period for the whole time.'
  * numberOfDaysInGroupForThePeriod
    - Returns: int
    - Function: Count number of days student was in the given group in the specified period of time'
  * getActiveGroupsAtAnyPointInThePeriod
    - Returns: List<String>
    - Function: Find the student's groups active at any point during specified period of time'
  * getActiveGroupsAtDate
    - Returns: List<String>
    - Function: Find the student's active groups at the given date'
  * getActiveGroupsDuringCompletePeriod
    - Returns: List<String>
    - Function: Find the student's groups active during whole specified period of time'
```

```
term
Purpose: The term of the funding element being evaluated.
Type: IAcademicTermPeriodAPI
Available Methods:
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getName
 - Returns: String
 - Function: No description available'
* isAttending
 - Returns: boolean
 - Function: No description available'
* isNotAttending
 - Returns: boolean
 - Function: No description available'
* isStandard
 - Returns: boolean
 - Function: No description available'
* isSummer
 - Returns: boolean
 - Function: No description available'
* isSummerHeader
 - Returns: boolean
 - Function: No description available'
* isSummerTrailer
 - Returns: boolean
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
termNo
Purpose: The number of the term of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
acy
Purpose: The academic year of the funding element being evaluated.
Type: IAcademicYearScheduleAPI
Available Methods:
* getCompletedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getPeriodOfEnrollmentEndDate
 - Returns: LocalDate
```

```
 - Function: No description available'
* getPeriodOfEnrollmentStartDate
 - Returns: LocalDate
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
acyNo
Purpose: The number of the academic year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
acyOrder
Purpose: The order number of the academic year of the funding element being evaluated, among all non-
canceled academic
years.
Type: Integer
Available Methods:
No methods available
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
```

```
    // Validation:
    // - [how boundaries are checked]
    // - [validation method details]
    // Edge Cases:
    // - [specific edge cases handled]
    // - [how each is managed]
    // Error Handling: [boundary violation management]
    // Impact: [effect of boundary violations]
    // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
   3. ARITHMETIC Section:
    Format:
    // ARITHMETIC:
    // Operation [number]: [name of calculation]
    // Formula: [complete mathematical expression]
    // Variables:
    // - [each variable and its purpose]
    // - [data types and units]
    // Steps:
    // 1. [detailed step-by-step breakdown]
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
   ADDITIONAL REQUIREMENTS:
   1. Complex Logic Sections:
    - Must break down multi-step operations
    - Explain interactions between components
    - Detail data flow and transformations
   2. Error Handling:
    - Document all error conditions
    - Explain recovery procedures
    - Detail logging and reporting
   3. API Interactions:
    - Document all platform method calls
    - Explain data transformations
    - Detail error handling
   4. Financial Aid Impact:
    - Explain effects on aid calculations
    - Document regulatory compliance
    - Detail student eligibility impact
   Example Section With All Aspects:
   // CONDITIONALS:
   // Check 1: Payment Period Status Validation
   // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
   // Purpose: Ensure only valid, active payment periods are included
   // Logic:
   // - Retrieve all loan payment periods
   // - Filter based on status (!= "CANCELED")
   // - Compare start dates with reference date
   // Success Path: Period included in calculations
   // Failure Path: Period excluded from totals
   // Business Rules: Canceled periods must not affect grade level
   // Error Handling: Invalid periods logged and skipped
   // Financial Impact: Affects available loan amounts
   //
   // BOUNDARY CHECKS:
   // Check 1: Payment Period Date Validation
   // Source: Academic Calendar Configuration
   // Variables: period.getStartDate(), referenceDate
   // Validation:
   // - Compare period start date with reference date
   // - Ensure dates are not null
   // Edge Cases:
```

```
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.PlusCreditDecisionMatchingCriteriaScript)
package CHANGE_ME // this should be updated to your actual package name
// Define utility methods
static BigDecimal nthIntPart(BigDecimal dividend, int divisor, int partIdx) {
 int truncatedDividend = dividend.intValue()
 BigDecimal fraction = dividend - truncatedDividend
 int roundedAmount = (dividend / divisor).intValue()
 return partIdx < truncatedDividend % divisor
 ? roundedAmount + 1
 : partIdx == divisor - 1
 ? roundedAmount + fraction
 : roundedAmount
}
// Actual script
boolean applicableLoansExist
if (program.isTerm()) {
 // Acy-scoped fund codes the client may need. Setting it as empty works too but in this case
 // the script can be significantly simplified. As this script only applies to (G)PLUS adding
 // fund codes with lower priority than (G)PLUS will make no difference.
 def acyScopedFundCodes = ["DISCOUNT"] as Set<String>
 def acyScopedTotal =
 financialPlan.getInAcademicYear(acyNo).filterFunds(acyScopedFundCodes.&contains).maxAmount
 def termsInNeed = acy.terms.indexed().findAll { tidx, t ->
 def termFundingSoFar = financialPlan.filterFunds { code -> !acyScopedFundCodes.contains(code) }
 .getOverlappingWith(t)
 .maxAmount
 def acyScopedInThisTerm = nthIntPart(acyScopedTotal, acy.terms.size(), tidx)
 def termCoa = coa.in(t).activeCoa
 def termRemainingNeed = [termCoa - termFundingSoFar - acyScopedInThisTerm, 0].max()
 termRemainingNeed > 0
 }.values()
 applicableLoansExist = termsInNeed.any { t ->
 !plusLoans.getOverlappingWith(t).isEmpty()
 }
} else {
 applicableLoansExist = acy.loanPaymentPeriods.any { pp ->
 pp.status != "CANCELED" && !plusLoans.getOverlappingWith(pp).isEmpty()
 }
}
if (!applicableLoansExist && plusOverrides.isEmpty() && deniedPlusLoans.isEmpty()) {
 return [projectedPlusLoan]
} else {
 return plusLoans.getOverlappingWith(pp).collect { loan ->
 loan.withNoPriority() // removing default priority queue here and rebuilding it below:
```
```

```
 .addReversedPriority(disbursements.getInScope(scope).isDisbursed())
 .addPriority(loan.getLatestPackagePriority(acyNo))
 .addPriority(loan.getProcessDate())
 .addReversedPriority(loan.getAcceptedLoanAmount())
 }
}
```
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Medical Amount

Name: medicalAmount

Purpose: Determine the medical amount. Script must return null (that is, not 0) to impart medical ineligibility.

In Baseline Config: No

Return Type: BigDecimal

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "medicalAmount"
Primary Function: Determine Medical amount. Script must return null (i.e. not 0) to impart Medical
 ineligibility.
Expected Output Type: BigDecimal
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
awy
Purpose: The award year of the funding element being evaluated.
```

```
Type: Integer
Available Methods:
No methods available
awyPeriod
Purpose: The period of the award year of the funding element being evaluated.
Type: IAwardYearAPI
Available Methods:
* asList
 - Returns: IAwardYearsAPI
 - Function: Returns an IAwardYearsAPI instance containing only this Award Year.'
* getEndYear
 - Returns: int
 - Function: Gets the end year of this Award Year, for example 2023 for Award Year 2022-2023.'
* getNext
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that follows this one chronologically.'
* getPrevious
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that precedes this one chronologically.'
* getStartYear
 - Returns: int
 - Function: Gets the start year of this Award Year, for example 2022 for Award Year 2022-2023.'
* getYear
 - Returns: int
 - Function: Short for getEndYear().'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
```

```
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
No methods available
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
```

```
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
```

```
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
```

```
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCode
Purpose: The fund code of the funding element being evaluated.
Type: String
Available Methods:
No methods available
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
```

```
 - Function: No description available'
fundType
Purpose: The fund type of the funding element being evaluated.
Type: String
Available Methods:
No methods available
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isFirstAcy
Purpose: True if the academic year of the funding element being evaluated is the first one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isLastAcy
Purpose: True if the academic year of the funding element being evaluated is the last one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
```

```
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
lp
Purpose: The loan period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
lpNo
Purpose: The number of the loan period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
pp
Purpose: The payment period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
ppNo
Purpose: The number of the payment period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
```

```
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
```

```
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
```

```
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
```

```
 - Function: Gets the status for a given period for the student.'
scope
Purpose: Gathers all the scope details of the script being evaluated (acyNo, fundCode,
fundType, awy, lpNo, ppNo) allowing quick access to financialPlan/disbursements/etc
subsets directly pertaining to this scope.
Type: IFundScopeAPI
Available Methods:
* getAcyNo
 - Returns: int
 - Function: No description available'
* getAwy
 - Returns: int
 - Function: No description available'
* getFundCode
 - Returns: String
 - Function: No description available'
* getFundType
 - Returns: String
 - Function: No description available'
* getLpNo
 - Returns: int
 - Function: No description available'
* getPpNo
 - Returns: int
 - Function: No description available'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
```

```
    - Returns: String
    - Function: Gets the first name.'
  * getGender
    - Returns: String
    - Function: Gets the gender.'
  * getHighSchoolEquivalency
    - Returns: Boolean
    - Function: Gets the high school equivalency status.'
  * getLastName
    - Returns: String
    - Function: Gets the last name.'
  * getMiddleName
    - Returns: String
    - Function: Gets the middle name.'
  * getPersonId
    - Returns: Long
    - Function: Gets the person ID.'
  * getProgramType
    - Returns: String
    - Function: Gets the program type.'
  * getRecentFinancialAidHistoryRecords
    - Returns: IRecentFahRecordsAPI
    - Function: Gets the recent financial aid records.'
  * getSchoolCode
    - Returns: String
    - Function: Gets the school code.'
  * getSchoolEmployee
    - Returns: Boolean
    - Function: Gets the school employee status.'
  * getSchoolEmployeeDependent
    - Returns: Boolean
    - Function: Gets the school employee dependent status.'
  * getSsn
    - Returns: ISocialSecurityNumber
    - Function: Gets the SSN.'
  * getStudentId
    - Returns: Long
    - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
Available Methods:
  * getPeriodsIntersectingWithGroupForThePeriod
    - Returns: List<? extends IPeriod>
    - Function: Find the periods student was in the given group in the specified period of time'
  * isInGroupAtAnyPointInThePeriod
    - Returns: boolean
    - Function: Checks if student was in the given group in the specified period at any point of time.'
  * isInGroupAtDate
    - Returns: boolean
    - Function: Checks if student was in the given group on the given date.'
  * isInGroupDuringCompletePeriod
    - Returns: boolean
    - Function: Checks if student was in the given group in the specified period for the whole time.'
  * numberOfDaysInGroupForThePeriod
    - Returns: int
    - Function: Count number of days student was in the given group in the specified period of time'
  * getActiveGroupsAtAnyPointInThePeriod
    - Returns: List<String>
    - Function: Find the student's groups active at any point during specified period of time'
  * getActiveGroupsAtDate
    - Returns: List<String>
    - Function: Find the student's active groups at the given date'
  * getActiveGroupsDuringCompletePeriod
    - Returns: List<String>
    - Function: Find the student's groups active during whole specified period of time'
```

```
term
Purpose: The term of the funding element being evaluated.
Type: IAcademicTermPeriodAPI
Available Methods:
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getName
 - Returns: String
 - Function: No description available'
* isAttending
 - Returns: boolean
 - Function: No description available'
* isNotAttending
 - Returns: boolean
 - Function: No description available'
* isStandard
 - Returns: boolean
 - Function: No description available'
* isSummer
 - Returns: boolean
 - Function: No description available'
* isSummerHeader
 - Returns: boolean
 - Function: No description available'
* isSummerTrailer
 - Returns: boolean
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
termNo
Purpose: The number of the term of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
acy
Purpose: The academic year of the funding element being evaluated.
Type: IAcademicYearScheduleAPI
Available Methods:
* getCompletedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getPeriodOfEnrollmentEndDate
 - Returns: LocalDate
```

```
 - Function: No description available'
* getPeriodOfEnrollmentStartDate
 - Returns: LocalDate
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
acyNo
Purpose: The number of the academic year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
acyOrder
Purpose: The order number of the academic year of the funding element being evaluated, among all non-
canceled academic
years.
Type: Integer
Available Methods:
No methods available
acys
Purpose: Provides access to all non-canceled academic years.
Type: IAcademicYearsScheduleAPI
Available Methods:
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
```

```
    // Variables: [list of variables and methods involved]
    // Purpose: [detailed business reason for this check]
    // Logic:
    // - [step-by-step breakdown of the condition]
    // - [how variables are used]
    // - [any transformations applied]
    // Success Path: [what happens when condition is true]
    // Failure Path: [what happens when condition is false]
    // Business Rules: [specific rules being enforced]
    // Error Handling: [how errors are managed]
    // Financial Impact: [effect on aid calculations]
    // If N/A, explicitly state: "CONDITIONALS: N/A"
  2. BOUNDARY CHECKS Section:
    Format:
    // BOUNDARY CHECKS:
    // Check [number]: [specific limit being enforced]
    // Source: [where boundary values come from]
    // Variables: [variables and methods involved]
    // Validation:
    // - [how boundaries are checked]
    // - [validation method details]
    // Edge Cases:
    // - [specific edge cases handled]
    // - [how each is managed]
    // Error Handling: [boundary violation management]
    // Impact: [effect of boundary violations]
    // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
  3. ARITHMETIC Section:
    Format:
    // ARITHMETIC:
    // Operation [number]: [name of calculation]
    // Formula: [complete mathematical expression]
    // Variables:
    // - [each variable and its purpose]
    // - [data types and units]
    // Steps:
    // 1. [detailed step-by-step breakdown]
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
  ADDITIONAL REQUIREMENTS:
  1. Complex Logic Sections:
    - Must break down multi-step operations
    - Explain interactions between components
    - Detail data flow and transformations
  2. Error Handling:
    - Document all error conditions
    - Explain recovery procedures
    - Detail logging and reporting
  3. API Interactions:
    - Document all platform method calls
    - Explain data transformations
    - Detail error handling
  4. Financial Aid Impact:
    - Explain effects on aid calculations
    - Document regulatory compliance
    - Detail student eligibility impact
  Example Section With All Aspects:
  // CONDITIONALS:
  // Check 1: Payment Period Status Validation
  // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
  // Purpose: Ensure only valid, active payment periods are included
```

```
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
N/A - There is no baseline config. This is a placeholder.
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Pell Enrollment Intensity

Name: pellEnrollmentIntensity

Purpose: Determine the enrollment intensity percentage for the PELL award.

In Baseline Config: Yes

Return Type: BigDecimal

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "pellEnrollmentIntensity"
Primary Function: Groovy script to determine the Enrollment Intensity percentage that should be applied to
 the calculated PELL award amount
Expected Output Type: BigDecimal
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
acy
Purpose: The academic year of the funding element being evaluated.
Type: IAcademicYearScheduleAPI
Available Methods:
* getCompletedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getPeriodOfEnrollmentEndDate
 - Returns: LocalDate
 - Function: No description available'
* getPeriodOfEnrollmentStartDate
 - Returns: LocalDate
 - Function: No description available'
* getTerms
```

```
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
acyNo
Purpose: The number of the academic year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
acyOrder
Purpose: The order number of the academic year of the funding element being evaluated, among all non-
canceled academic
years.
Type: Integer
Available Methods:
No methods available
acys
Purpose: Provides access to all non-canceled academic years.
Type: IAcademicYearsScheduleAPI
Available Methods:
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
awy
Purpose: The award year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
awyPeriod
Purpose: The period of the award year of the funding element being evaluated.
Type: IAwardYearAPI
Available Methods:
* asList
 - Returns: IAwardYearsAPI
 - Function: Returns an IAwardYearsAPI instance containing only this Award Year.'
* getEndYear
 - Returns: int
 - Function: Gets the end year of this Award Year, for example 2023 for Award Year 2022-2023.'
* getNext
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that follows this one chronologically.'
* getPrevious
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that precedes this one chronologically.'
* getStartYear
 - Returns: int
 - Function: Gets the start year of this Award Year, for example 2022 for Award Year 2022-2023.'
* getYear
 - Returns: int
 - Function: Short for getEndYear().'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
```

```
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
No methods available
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
```

```
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
```

```
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
financialPlan
```

```
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
```

```
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCode
Purpose: The fund code of the funding element being evaluated.
Type: String
Available Methods:
No methods available
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
fundType
Purpose: The fund type of the funding element being evaluated.
Type: String
Available Methods:
No methods available
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
```

```
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isFirstAcy
Purpose: True if the academic year of the funding element being evaluated is the first one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isLastAcy
Purpose: True if the academic year of the funding element being evaluated is the last one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
lp
Purpose: The loan period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
```

```
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
lpNo
Purpose: The number of the loan period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
pp
Purpose: The payment period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
ppNo
Purpose: The number of the payment period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
```

```
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
```

```
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
```

```
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
scope
Purpose: Gathers all the scope details of the script being evaluated (acyNo, fundCode,
fundType, awy, lpNo, ppNo) allowing quick access to financialPlan/disbursements/etc
subsets directly pertaining to this scope.
Type: IFundScopeAPI
Available Methods:
* getAcyNo
 - Returns: int
 - Function: No description available'
* getAwy
 - Returns: int
 - Function: No description available'
* getFundCode
 - Returns: String
 - Function: No description available'
* getFundType
 - Returns: String
 - Function: No description available'
* getLpNo
 - Returns: int
 - Function: No description available'
* getPpNo
 - Returns: int
 - Function: No description available'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
```

```
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
```

```
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
Available Methods:
* getPeriodsIntersectingWithGroupForThePeriod
 - Returns: List<? extends IPeriod>
 - Function: Find the periods student was in the given group in the specified period of time'
* isInGroupAtAnyPointInThePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period at any point of time.'
* isInGroupAtDate
 - Returns: boolean
 - Function: Checks if student was in the given group on the given date.'
* isInGroupDuringCompletePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period for the whole time.'
* numberOfDaysInGroupForThePeriod
 - Returns: int
 - Function: Count number of days student was in the given group in the specified period of time'
* getActiveGroupsAtAnyPointInThePeriod
 - Returns: List<String>
 - Function: Find the student's groups active at any point during specified period of time'
* getActiveGroupsAtDate
 - Returns: List<String>
 - Function: Find the student's active groups at the given date'
* getActiveGroupsDuringCompletePeriod
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
term
Purpose: The term of the funding element being evaluated.
Type: IAcademicTermPeriodAPI
Available Methods:
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getName
 - Returns: String
 - Function: No description available'
* isAttending
 - Returns: boolean
 - Function: No description available'
* isNotAttending
 - Returns: boolean
 - Function: No description available'
* isStandard
 - Returns: boolean
 - Function: No description available'
* isSummer
 - Returns: boolean
 - Function: No description available'
* isSummerHeader
 - Returns: boolean
 - Function: No description available'
* isSummerTrailer
```

```
 - Returns: boolean
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
termNo
Purpose: The number of the term of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
```

```
    // Operation [number]: [name of calculation]
    // Formula: [complete mathematical expression]
    // Variables:
    // - [each variable and its purpose]
    // - [data types and units]
    // Steps:
    // 1. [detailed step-by-step breakdown]
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
   ADDITIONAL REQUIREMENTS:
   1. Complex Logic Sections:
    - Must break down multi-step operations
    - Explain interactions between components
    - Detail data flow and transformations
   2. Error Handling:
    - Document all error conditions
    - Explain recovery procedures
    - Detail logging and reporting
   3. API Interactions:
    - Document all platform method calls
    - Explain data transformations
    - Detail error handling
   4. Financial Aid Impact:
    - Explain effects on aid calculations
    - Document regulatory compliance
    - Detail student eligibility impact
   Example Section With All Aspects:
   // CONDITIONALS:
   // Check 1: Payment Period Status Validation
   // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
   // Purpose: Ensure only valid, active payment periods are included
   // Logic:
   // - Retrieve all loan payment periods
   // - Filter based on status (!= "CANCELED")
   // - Compare start dates with reference date
   // Success Path: Period included in calculations
   // Failure Path: Period excluded from totals
   // Business Rules: Canceled periods must not affect grade level
   // Error Handling: Invalid periods logged and skipped
   // Financial Impact: Affects available loan amounts
   //
   // BOUNDARY CHECKS:
   // Check 1: Payment Period Date Validation
   // Source: Academic Calendar Configuration
   // Variables: period.getStartDate(), referenceDate
   // Validation:
   // - Compare period start date with reference date
   // - Ensure dates are not null
   // Edge Cases:
   // - Null dates are excluded
   // - Future dates are filtered
   // Error Handling: Invalid dates logged and skipped
   // Impact: Prevents future periods from affecting current calculation
   //
   // ARITHMETIC:
   // Operation 1: Payment Period Units Summation
   // Formula: sum(validPeriods.map(period => period.getUnits()))
   // Variables:
   // - period.getUnits(): Decimal, represents credit hours
   // - validPeriods: Array of filtered payment periods
   // Steps:
```

```
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.PellEnrollmentIntensityScript)
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
import org.joda.time.LocalDate
def countedUnits = 0.0
def enrollmentIntensity = 0.0
def creditsRequiredForFullTimeEnrollment = 12
LocalDate today = LocalDate.now()
LocalDate censusDate
// if enrollment status is X return 0 (0%)
if (primaryProgram.getEnrollmentStatus() != null && primaryProgram.getEnrollmentStatus() == "X") {
 return 0
}
// if not term, return 1 (100%)
if (!program.isTerm()) {
 return 1
}
//term program logic
censusDate = term.getStartDate().plusDays(10)
//count credits and calculate enrollmentIntensity
for (def course: program.getCourses().getAssociatedTo(term)) {
 log.debug("PellEnrollmentIntensity script, COURSE IN LOOP: courseEndDate = {}, courseAraIndicator = {},
 courseSchedulingStatus = {}, courseUnitsToAdd = {}", course.getEndDate(), course.getFirstAraIndicator(),
 course.getSchedulingStatus(), course.getUnits())
 // If the course is withdrawn, only count the credits if the course was withdrawn after the census date. If
 the course is not withdrawn, count the course credits.
 if ((["WITHDRAWN"].contains(course.getSchedulingStatus()) && course.getLastDateOfAttendance() != null &&
 course.getLastDateOfAttendance() > censusDate) || course.getSchedulingStatus() != "WITHDRAWN") {
 countedUnits = countedUnits + course.getUnits()
 }
 enrollmentIntensity = countedUnits / creditsRequiredForFullTimeEnrollment
}
//census date logic
if (censusDate != null && today <= censusDate) {
 log.debug("PellEnrollmentIntensity script if statement final values: enrollmentIntensity = {},
 countedUnits = {}, creditsRequiredForFullTimeEnrollment = {}", enrollmentIntensity, countedUnits,
 creditsRequiredForFullTimeEnrollment)
 return enrollmentIntensity
}
else {
 //define previousFpo
 log.debug("PellEnrollmentIntensity script else statement begin")
 def previousFpo
 if (previousFinancialPlan.effectiveAt(censusDate) != null) {
 log.debug("PellEnrollmentIntensity script using census date previous FPO")
 previousFpo = previousFinancialPlan.effectiveAt(censusDate)
 } else if (financialPlan.getVersion() > 1) {
 log.debug("PellEnrollmentIntensity script using (FPO version - 1) previous FPO")
 previousFpo = previousFinancialPlan.withVersion(financialPlan.getVersion() - 1)
 }
 //previousFpo is not null logic
 if (previousFpo != null) {
 log.debug("PellEnrollmentIntensity script previousFpo is not null")
 def prevEnrollmentIntensities =
 previousFpo.getInAwardYear(awy).getOverlappingWith(term).getEnrollmentIntensities()
```
```

```
 if (prevEnrollmentIntensities.isEmpty()) {
 log.debug("PellEnrollmentIntensity script, prevEnrollmentIntensities is empty, final values:
 enrollmentIntensity = {}, countedUnits = {}, creditsRequiredForFullTimeEnrollment = {}",
 enrollmentIntensity, countedUnits, creditsRequiredForFullTimeEnrollment)
 return enrollmentIntensity
 }
 enrollmentIntensity = prevEnrollmentIntensities.get()
 log.debug("PellEnrollmentIntensity script, prevEnrollmentIntensities is NOT empty, final
 values: enrollmentIntensity = {}, countedUnits = {}, creditsRequiredForFullTimeEnrollment = {}",
 enrollmentIntensity, countedUnits, creditsRequiredForFullTimeEnrollment)
 }
 else {
 //previousFpo is null message
 log.debug("PellEnrollmentIntensity script previousFpo is null")
 }
}
return enrollmentIntensity
```
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Enrollment and Academic Scripts

These are the prompts for scripts managing enrollment status, term definitions, and academic year calculations.

## Enrollment Status Override

Name: enrollmentStatusOverride

Purpose: Determines whether the calculated enrollment status should be overridden.

In Baseline Config: Yes

Return Type: String

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "enrollmentStatusOverride"
```

```
Primary Function: Groovy script to determine whether the calculated enrollment status (as defined in the
 School CSV's Enrollment
Status Determination column) should be overridden. Example usages include freezing enrollment status as of a
certain census date.
Expected Output Type: String
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
acy
Purpose: The academic year of the funding element being evaluated.
Type: IAcademicYearScheduleAPI
Available Methods:
* getCompletedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getPeriodOfEnrollmentEndDate
 - Returns: LocalDate
 - Function: No description available'
* getPeriodOfEnrollmentStartDate
 - Returns: LocalDate
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
acyNo
Purpose: The number of the academic year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
acyOrder
Purpose: The order number of the academic year of the funding element being evaluated, among all non-
canceled academic
```

```
years.
Type: Integer
Available Methods:
No methods available
acys
Purpose: Provides access to all non-canceled academic years.
Type: IAcademicYearsScheduleAPI
Available Methods:
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
awy
Purpose: The award year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
awyPeriod
Purpose: The period of the award year of the funding element being evaluated.
Type: IAwardYearAPI
Available Methods:
* asList
 - Returns: IAwardYearsAPI
 - Function: Returns an IAwardYearsAPI instance containing only this Award Year.'
* getEndYear
 - Returns: int
 - Function: Gets the end year of this Award Year, for example 2023 for Award Year 2022-2023.'
* getNext
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that follows this one chronologically.'
* getPrevious
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that precedes this one chronologically.'
* getStartYear
 - Returns: int
 - Function: Gets the start year of this Award Year, for example 2022 for Award Year 2022-2023.'
* getYear
 - Returns: int
 - Function: Short for getEndYear().'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
```

```
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
No methods available
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
```

```
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
```

```
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollmentStatus
Purpose: The student's enrollment status. Can be one of FULL_TIME, THREE_QUARTER_TIME, HALF_TIME,
LESS_THAN_HALF_TIME or NOT_ATTENDING.
Type: String
Available Methods:
No methods available
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
```

```
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
```

```
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCode
Purpose: The fund code of the funding element being evaluated.
Type: String
Available Methods:
No methods available
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
fundType
Purpose: The fund type of the funding element being evaluated.
Type: String
Available Methods:
No methods available
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
```

```
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isFirstAcy
Purpose: True if the academic year of the funding element being evaluated is the first one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isLastAcy
Purpose: True if the academic year of the funding element being evaluated is the last one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
lp
Purpose: The loan period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
lpNo
Purpose: The number of the loan period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
```

```
period
Purpose: A period matching the payment period being evaluated.
Type: IPeriod
Available Methods:
No methods available
pp
Purpose: The payment period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
ppNo
Purpose: The number of the payment period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
```

```
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
```

```
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
```

```
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
sapStatus
Purpose: The student's SAP status. Can be one of RG_REGULAR, FW_FINANCIAL_AID_WARNING,
FD_FINANCIAL_AID_DISQUALIFICATION or FP_FINANCIAL_AID_PROBATION.
Type: String
Available Methods:
No methods available
scope
Purpose: Gathers all the scope details of the script being evaluated (acyNo, fundCode,
fundType, awy, lpNo, ppNo) allowing quick access to financialPlan/disbursements/etc
subsets directly pertaining to this scope.
Type: IFundScopeAPI
Available Methods:
* getAcyNo
 - Returns: int
 - Function: No description available'
* getAwy
 - Returns: int
 - Function: No description available'
* getFundCode
 - Returns: String
 - Function: No description available'
* getFundType
 - Returns: String
 - Function: No description available'
* getLpNo
 - Returns: int
 - Function: No description available'
* getPpNo
 - Returns: int
 - Function: No description available'
student
```

```
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
```

```
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
Available Methods:
* getPeriodsIntersectingWithGroupForThePeriod
 - Returns: List<? extends IPeriod>
 - Function: Find the periods student was in the given group in the specified period of time'
* isInGroupAtAnyPointInThePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period at any point of time.'
* isInGroupAtDate
 - Returns: boolean
 - Function: Checks if student was in the given group on the given date.'
* isInGroupDuringCompletePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period for the whole time.'
* numberOfDaysInGroupForThePeriod
 - Returns: int
 - Function: Count number of days student was in the given group in the specified period of time'
* getActiveGroupsAtAnyPointInThePeriod
 - Returns: List<String>
 - Function: Find the student's groups active at any point during specified period of time'
* getActiveGroupsAtDate
 - Returns: List<String>
 - Function: Find the student's active groups at the given date'
* getActiveGroupsDuringCompletePeriod
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
term
Purpose: The term of the funding element being evaluated.
Type: IAcademicTermPeriodAPI
Available Methods:
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getName
 - Returns: String
 - Function: No description available'
* isAttending
 - Returns: boolean
 - Function: No description available'
* isNotAttending
 - Returns: boolean
 - Function: No description available'
* isStandard
 - Returns: boolean
 - Function: No description available'
* isSummer
 - Returns: boolean
 - Function: No description available'
* isSummerHeader
```

```
 - Returns: boolean
 - Function: No description available'
* isSummerTrailer
 - Returns: boolean
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
termNo
Purpose: The number of the term of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
```

```
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
 // Operation [number]: [name of calculation]
 // Formula: [complete mathematical expression]
 // Variables:
 // - [each variable and its purpose]
 // - [data types and units]
 // Steps:
 // 1. [detailed step-by-step breakdown]
 // 2. [intermediate calculations]
 // 3. [transformations applied]
 // Business Context: [why this calculation is needed]
 // Result: [meaning of the output]
 // Financial Impact: [effect on aid calculations]
 // Limitations: [any mathematical constraints]
 // If N/A, explicitly state: "ARITHMETIC: N/A"
ADDITIONAL REQUIREMENTS:
1. Complex Logic Sections:
 - Must break down multi-step operations
 - Explain interactions between components
 - Detail data flow and transformations
2. Error Handling:
 - Document all error conditions
 - Explain recovery procedures
 - Detail logging and reporting
3. API Interactions:
 - Document all platform method calls
 - Explain data transformations
 - Detail error handling
4. Financial Aid Impact:
 - Explain effects on aid calculations
 - Document regulatory compliance
 - Detail student eligibility impact
Example Section With All Aspects:
// CONDITIONALS:
// Check 1: Payment Period Status Validation
// Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
```

```
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.EnrollmentStatusOverrideScript)
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
/* Pell Enrollment Status Override with Census Date - combined Term and Non-Term*/
import org.joda.time.LocalDate;
if (program.isTerm())
{
 LocalDate termStartDate = term.getStartDate();
 LocalDate termEndDate = term.getEndDate();
 LocalDate censusDate;
 LocalDate today = LocalDate.now();
 String programType = primaryProgram.getProgramType();
 censusDate = term.getStartDate().plusDays(10);
 log.debug("CONFIG LOGGING STARTS HERE");
 log.debug("censusDate = {}", censusDate);
 log.debug("today = {}", today);
 log.debug("termStartDate = {}", termStartDate);
 def safiTerm = program.getTerms().getOverlappingWith(term).get(0);
 if (censusDate != null && today > censusDate && (program.getEnrollmentStatus() == "W" ||
 safiTerm.getStudentsTermStatus() == "WITHDRAWN"))
 {
 return enrollmentStatus;
 /*
 def countedUnits = 0.0;
 for (def course: program.getCourses().getAssociatedTo(period))
 {
 if (["WITHDRAWN"].contains(course.getSchedulingStatus()))
 {
 if (course.getLastDateOfAttendance() != null && ((program.getOfficialLastDateOfAttendance() != null &&
 course.getLastDateOfAttendance() == program.getOfficialLastDateOfAttendance()) || (safiTerm.getOlda() !=
 null && course.getLastDateOfAttendance() == safiTerm.getOlda())))
 {
 log.debug("GLOBAL ENROLLMENT SCRIPT - WITHDRAWN: courseLDA = {}, courseSchedulingStatus = {},
 programOLDA = {}, courseUnitsToAdd = {}", course.getLastDateOfAttendance(), course.getSchedulingStatus(),
 program.getOfficialLastDateOfAttendance(), course.getUnits());
 countedUnits = countedUnits + course.getUnits();
 }
 }
 else if (["SCHEDULED", "PASSED", "FAILED"].contains(course.getSchedulingStatus()))
 {
 log.debug("GLOBAL ENROLLMENT SCRIPT - SCHEDULED PASSED FAILED: courseSchedulingStatus = {},
 courseUnitsToAdd = {}", course.getSchedulingStatus(), course.getUnits());
 countedUnits = countedUnits + course.getUnits();
 }
 }
 def NA = 0.0;
 def LTHT = 6.0;
 def HT = 9.0;
 def TQT = 12.0;
 return helper.mapRanges()
 .le(NA, "NOT_ATTENDING")
 .lt(LTHT, "LESS_THAN_HALF_TIME")
```
```

```
.lt(HT, "HALF_TIME")
.lt(TQT, "THREE_QUARTER_TIME")
.defaultValue("FULL_TIME")
.apply(countedUnits);
*/
}
else if (censusDate != null && today > censusDate)
{
log.debug("FIRST IF PASSED");
def previousFpo;
if (previousFinancialPlan.effectiveAt(censusDate) != null)
{
log.debug("USING CENSUS DATE PREVIOUS FPO");
previousFpo = previousFinancialPlan.effectiveAt(censusDate);
}
else if (financialPlan.getVersion() > 1)
{
log.debug("USING FPO VERSION - 1 PREVIOUS FPO");
previousFpo = previousFinancialPlan.withVersion(financialPlan.getVersion() - 1);
}
if (previousFpo != null)
{
def pellFundStatuses = previousFpo.getFund("PELL").getInAwardYear(awy).getFundStatuses();
if (!pellFundStatuses.contains("AWARDED") && !pellFundStatuses.contains("ESTIMATED"))
{
return enrollmentStatus;
}
log.debug("PREVIOUS FPO IS NOT NULL");
def previousPell = previousFpo.getFund("PELL").getInAcademicYear(acyNo).getInPaymentPeriod(ppNo);
if (previousPell.isEligible())
{
log.debug("PREVIOUS PELL IS ELIGIBLE");
if (!previousPell.getEnrollmentStatuses().isEmpty())
{
log.debug("PREVIOUS PELL HAS NON NULL ENROLLMENT STATUS");
log.debug("PREVIOUS PELL ENROLLMENT STATUS = {}", previousPell.getEnrollmentStatuses().get());
return previousPell.getEnrollmentStatuses().get();
}
}
}
}
return enrollmentStatus;
}
return enrollmentStatus;
```
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Global Enrollment Status Criteria

Name: globalEnrollmentStatusCriteria

Purpose: Specifies how the application should find out the student's enrollment status.

In Baseline Config: Yes

Return Type: String

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "globalEnrollmentStatusCriteria"
Primary Function: Specifies how the system should determine the student's enrollment status.
Expected Output Type: String
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
No methods available
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
```

```
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
```

```
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
```

```
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
```

```
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
```

```
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
```

```
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
```

```
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
safiCreatedDate
Purpose: Date when the latest SAFI was created.
Type: LocalDate
Available Methods:
No methods available
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
```

```
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
Available Methods:
* getPeriodsIntersectingWithGroupForThePeriod
 - Returns: List<? extends IPeriod>
 - Function: Find the periods student was in the given group in the specified period of time'
* isInGroupAtAnyPointInThePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period at any point of time.'
* isInGroupAtDate
 - Returns: boolean
 - Function: Checks if student was in the given group on the given date.'
* isInGroupDuringCompletePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period for the whole time.'
* numberOfDaysInGroupForThePeriod
 - Returns: int
 - Function: Count number of days student was in the given group in the specified period of time'
* getActiveGroupsAtAnyPointInThePeriod
```

```
 - Returns: List<String>
 - Function: Find the student's groups active at any point during specified period of time'
* getActiveGroupsAtDate
 - Returns: List<String>
 - Function: Find the student's active groups at the given date'
* getActiveGroupsDuringCompletePeriod
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
term
Purpose: A period matching the term being evaluated.
Type: ICourseAssociablePeriodAPI
Available Methods:
* isAssociatedTo
 - Returns: boolean
 - Function: Checks if a course is associated with this period.'
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
```

```
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
 // Operation [number]: [name of calculation]
 // Formula: [complete mathematical expression]
 // Variables:
 // - [each variable and its purpose]
 // - [data types and units]
 // Steps:
 // 1. [detailed step-by-step breakdown]
 // 2. [intermediate calculations]
 // 3. [transformations applied]
 // Business Context: [why this calculation is needed]
 // Result: [meaning of the output]
 // Financial Impact: [effect on aid calculations]
 // Limitations: [any mathematical constraints]
 // If N/A, explicitly state: "ARITHMETIC: N/A"
ADDITIONAL REQUIREMENTS:
1. Complex Logic Sections:
 - Must break down multi-step operations
 - Explain interactions between components
 - Detail data flow and transformations
2. Error Handling:
 - Document all error conditions
 - Explain recovery procedures
 - Detail logging and reporting
3. API Interactions:
 - Document all platform method calls
 - Explain data transformations
 - Detail error handling
4. Financial Aid Impact:
 - Explain effects on aid calculations
 - Document regulatory compliance
 - Detail student eligibility impact
```

```
Example Section With All Aspects:
// CONDITIONALS:
// Check 1: Payment Period Status Validation
// Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.GlobalEnrollmentStatusCriteriaScript)
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
/* Global Enrollment Status - SFP Baseline - Combined Term and Non-Term */
/*
def actualCourses = helper.filterCourses(program)
 .period(term)
 .safiCreatedDate(safiCreatedDate)
 .go();
*/
import org.joda.time.LocalDate;
if (program.isTerm())
{
 // If the student decides to cancel their enrollment and the Program Enrollment Status is X, then return
 NOT_ATTENDING
 if (primaryProgram.getEnrollmentStatus() != null && primaryProgram.getEnrollmentStatus() == "X")
 {
 return "NOT_ATTENDING";
 }
 def countedUnits = 0.0;
```
```

```
def safiTerm = program.getTerms().getOverlappingWith(term).get(0);
// Institutions should define their own add drop policy
def addDropPolicy = term.getStartDate().plusDays(10);
log.debug ("Term start date = {}, term start date + 10 = {}", term.getStartDate(), addDropPolicy);
for (def course: program.getCourses().getAssociatedTo(period))
{
log.debug("GLOBAL ENROLLMENT SCRIPT, COURSE IN LOOP: courseEndDate = {}, courseAraIndicator = {},
courseSchedulingStatus = {}, courseUnitsToAdd = {}", course.getEndDate(), course.getFirstAraIndicator(),
course.getSchedulingStatus(), course.getUnits());
// If the course is withdrawn, only count the credits if the course was withdrawn after the add/drop
period. If the course is not withdrawn, count the course credits.
if ((["WITHDRAWN"].contains(course.getSchedulingStatus()) && course.getLastDateOfAttendance() != null &&
course.getLastDateOfAttendance() > addDropPolicy) || course.getSchedulingStatus() != "WITHDRAWN")
{
countedUnits = countedUnits + course.getUnits();
}
}
def NA = 0.0;
def LTHT = 6.0;
def HT = 9.0;
def TQT = 12.0;
return helper.mapRanges()
.le(NA, "NOT_ATTENDING")
.lt(LTHT, "LESS_THAN_HALF_TIME")
.lt(HT, "HALF_TIME")
.lt(TQT, "THREE_QUARTER_TIME")
.defaultValue("FULL_TIME")
.apply(countedUnits);
}
return "FULL_TIME";
```

```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Summer Term Type

Name: summerTermType

Purpose: Defines if the Summer Term should be treated as Header or Trailer.

In Baseline Config: No

Return Type: String

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
```

```
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "summerTermType"
Primary Function: Defines if the Summer Term should be treated as a Header or Trailer.
Expected Output Type: String
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
groovyScriptHelper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
primaryProgram
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
```

```
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
```

```
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
term
Purpose: A period matching the term being evaluated.
Type: IPeriod
Available Methods:
No methods available
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
```

```
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
 // Operation [number]: [name of calculation]
 // Formula: [complete mathematical expression]
 // Variables:
 // - [each variable and its purpose]
 // - [data types and units]
 // Steps:
 // 1. [detailed step-by-step breakdown]
 // 2. [intermediate calculations]
 // 3. [transformations applied]
 // Business Context: [why this calculation is needed]
 // Result: [meaning of the output]
 // Financial Impact: [effect on aid calculations]
 // Limitations: [any mathematical constraints]
 // If N/A, explicitly state: "ARITHMETIC: N/A"
ADDITIONAL REQUIREMENTS:
```

```
1. Complex Logic Sections:
 - Must break down multi-step operations
 - Explain interactions between components
 - Detail data flow and transformations
2. Error Handling:
 - Document all error conditions
 - Explain recovery procedures
 - Detail logging and reporting
3. API Interactions:
 - Document all platform method calls
 - Explain data transformations
 - Detail error handling
4. Financial Aid Impact:
 - Explain effects on aid calculations
 - Document regulatory compliance
 - Detail student eligibility impact
Example Section With All Aspects:
// CONDITIONALS:
// Check 1: Payment Period Status Validation
// Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
N/A - There is no baseline config. This is a placeholder.
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
```

```
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Nonterm ACY Months

Name: nonTermAcyMonths

Purpose: Determine the number of months in a nonterm academic year.

In Baseline Config: Yes

Return Type: Integer

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "nonTermAcyMonths"
Primary Function: Determine the number of months in a non-term academic year.
Expected Output Type: Integer
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
acy
Purpose: The academic year of the funding element being evaluated.
Type: IAcademicYearScheduleAPI
Available Methods:
* getCompletedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getGrantPaymentPeriods
```

```
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getPeriodOfEnrollmentEndDate
 - Returns: LocalDate
 - Function: No description available'
* getPeriodOfEnrollmentStartDate
 - Returns: LocalDate
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
acyNo
Purpose: The number of the academic year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
acyOrder
Purpose: The order number of the academic year of the funding element being evaluated, among all non-
canceled academic
years.
Type: Integer
Available Methods:
No methods available
acys
Purpose: Provides access to all non-canceled academic years.
Type: IAcademicYearsScheduleAPI
Available Methods:
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
awy
Purpose: The award year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
awyPeriod
Purpose: The period of the award year of the funding element being evaluated.
Type: IAwardYearAPI
Available Methods:
* asList
 - Returns: IAwardYearsAPI
 - Function: Returns an IAwardYearsAPI instance containing only this Award Year.'
* getEndYear
 - Returns: int
 - Function: Gets the end year of this Award Year, for example 2023 for Award Year 2022-2023.'
* getNext
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that follows this one chronologically.'
```

```
* getPrevious
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that precedes this one chronologically.'
* getStartYear
 - Returns: int
 - Function: Gets the start year of this Award Year, for example 2022 for Award Year 2022-2023.'
* getYear
 - Returns: int
 - Function: Short for getEndYear().'
fundCode
Purpose: The fund code of the funding element being evaluated.
Type: String
Available Methods:
No methods available
fundType
Purpose: The fund type of the funding element being evaluated.
Type: String
Available Methods:
No methods available
groovyScriptHelper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isFirstAcy
Purpose: True if the academic year of the funding element being evaluated is the first one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isLastAcy
Purpose: True if the academic year of the funding element being evaluated is the last one among all non-
canceled academic
```

```
years.
Type: Boolean
Available Methods:
No methods available
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
lp
Purpose: The loan period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
lpNo
Purpose: The number of the loan period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
pp
Purpose: The payment period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
ppNo
Purpose: The number of the payment period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
```

```
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
```

```
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
scope
Purpose: Gathers all the scope details of the script being evaluated (acyNo, fundCode,
fundType, awy, lpNo, ppNo) allowing quick access to financialPlan/disbursements/etc
subsets directly pertaining to this scope.
Type: IFundScopeAPI
Available Methods:
* getAcyNo
 - Returns: int
 - Function: No description available'
* getAwy
 - Returns: int
 - Function: No description available'
* getFundCode
 - Returns: String
 - Function: No description available'
* getFundType
 - Returns: String
 - Function: No description available'
```

```
* getLpNo
 - Returns: int
 - Function: No description available'
* getPpNo
 - Returns: int
 - Function: No description available'
term
Purpose: The term of the funding element being evaluated.
Type: IAcademicTermPeriodAPI
Available Methods:
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getName
 - Returns: String
 - Function: No description available'
* isAttending
 - Returns: boolean
 - Function: No description available'
* isNotAttending
 - Returns: boolean
 - Function: No description available'
* isStandard
 - Returns: boolean
 - Function: No description available'
* isSummer
 - Returns: boolean
 - Function: No description available'
* isSummerHeader
 - Returns: boolean
 - Function: No description available'
* isSummerTrailer
 - Returns: boolean
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
termNo
Purpose: The number of the term of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
```

```
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
 // Operation [number]: [name of calculation]
 // Formula: [complete mathematical expression]
 // Variables:
 // - [each variable and its purpose]
 // - [data types and units]
 // Steps:
 // 1. [detailed step-by-step breakdown]
 // 2. [intermediate calculations]
 // 3. [transformations applied]
 // Business Context: [why this calculation is needed]
 // Result: [meaning of the output]
 // Financial Impact: [effect on aid calculations]
 // Limitations: [any mathematical constraints]
 // If N/A, explicitly state: "ARITHMETIC: N/A"
ADDITIONAL REQUIREMENTS:
1. Complex Logic Sections:
 - Must break down multi-step operations
 - Explain interactions between components
 - Detail data flow and transformations
2. Error Handling:
 - Document all error conditions
 - Explain recovery procedures
 - Detail logging and reporting
3. API Interactions:
 - Document all platform method calls
 - Explain data transformations
 - Detail error handling
4. Financial Aid Impact:
 - Explain effects on aid calculations
 - Document regulatory compliance
 - Detail student eligibility impact
```

```
Example Section With All Aspects:
// CONDITIONALS:
// Check 1: Payment Period Status Validation
// Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.NonTermAcyMonthsScript)
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
/* Non-Term Baseline ACY Months Calculation, return type int */
if (acy.getUnits() == null || program.getAydUnits() == null)
{
 log.debug("ACY MONTHS CALCULATION CONFIG: ERROR, null values passed from APIs");
 return 0;
}
log.debug("ACY MONTHS CALCULATION CONFIG: Academic Year Schedule Units = {}, Program AYD Units = {}",
 acy.getUnits(), program.getAydUnits());
if ((acy.getUnits() / program.getAydUnits()) >= 1)
{
 return 9;
}
else
{
 int returnValue = Math.round(((acy.getUnits()/program.getAydUnits()) * 9));
 return returnValue;
}
```
```

```
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Term Payment Period Months

Name: termPaymentPeriodMonths

Purpose: Determine the number of months in a term payment period.

In Baseline Config: Yes

Return Type: BigDecimal

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "termPaymentPeriodMonths"
Primary Function: Determine the number of months in a term payment period.
Expected Output Type: BigDecimal
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
```
```

```
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
```

```
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
```

```
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
term
Purpose: A period matching the payment period being evaluated.
Type: IPeriod
Available Methods:
No methods available
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
```

```
   - Critical business rules enforced
   - Boundary conditions and limits
   - Financial aid implications
   - Required inputs and expected outputs
   - Error handling approach
   - Integration points
   - Compliance considerations
  SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
  Every logical code section must include detailed comments for ALL three aspects:
  1. CONDITIONALS Section:
   Format:
   // CONDITIONALS:
   // Check [number]: [specific condition being evaluated]
   // Variables: [list of variables and methods involved]
   // Purpose: [detailed business reason for this check]
   // Logic:
   // - [step-by-step breakdown of the condition]
   // - [how variables are used]
   // - [any transformations applied]
   // Success Path: [what happens when condition is true]
   // Failure Path: [what happens when condition is false]
   // Business Rules: [specific rules being enforced]
   // Error Handling: [how errors are managed]
   // Financial Impact: [effect on aid calculations]
   // If N/A, explicitly state: "CONDITIONALS: N/A"
  2. BOUNDARY CHECKS Section:
   Format:
   // BOUNDARY CHECKS:
   // Check [number]: [specific limit being enforced]
   // Source: [where boundary values come from]
   // Variables: [variables and methods involved]
   // Validation:
   // - [how boundaries are checked]
   // - [validation method details]
   // Edge Cases:
   // - [specific edge cases handled]
   // - [how each is managed]
   // Error Handling: [boundary violation management]
   // Impact: [effect of boundary violations]
   // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
  3. ARITHMETIC Section:
   Format:
   // ARITHMETIC:
   // Operation [number]: [name of calculation]
   // Formula: [complete mathematical expression]
   // Variables:
   // - [each variable and its purpose]
   // - [data types and units]
   // Steps:
   // 1. [detailed step-by-step breakdown]
   // 2. [intermediate calculations]
   // 3. [transformations applied]
   // Business Context: [why this calculation is needed]
   // Result: [meaning of the output]
   // Financial Impact: [effect on aid calculations]
   // Limitations: [any mathematical constraints]
   // If N/A, explicitly state: "ARITHMETIC: N/A"
  ADDITIONAL REQUIREMENTS:
  1. Complex Logic Sections:
   - Must break down multi-step operations
   - Explain interactions between components
   - Detail data flow and transformations
  2. Error Handling:
   - Document all error conditions
   - Explain recovery procedures
   - Detail logging and reporting
```

```
3. API Interactions:
 - Document all platform method calls
 - Explain data transformations
 - Detail error handling
4. Financial Aid Impact:
 - Explain effects on aid calculations
 - Document regulatory compliance
 - Detail student eligibility impact
Example Section With All Aspects:
// CONDITIONALS:
// Check 1: Payment Period Status Validation
// Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.TermPaymentPeriodMonthsScript)
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
/* Term Baseline PP Months Calculation, return type BigDecimal */
import org.joda.time.LocalDate;
def returnValue = 1.0;
def programTermType = program.getTermType();
log.debug("PP MONTHS CALCULATION CONFIG: Program Term Type = {}", program.getTermType());
def safiTerm = program.getTerms().getOverlappingWith(term).get(0);
if (program.getTermType() == null)
{
 log.debug("PP MONTHS CALCULATION CONFIG: ERROR, null value passed from API");
```
```

```
 returnValue = 1.0;
}
else if (safiTerm.isSummer())
{
 returnValue = 2.0;
}
// Standard from this point on
else if (programTermType == "Semester")
{
 returnValue = 4.5;
}
else if (programTermType == "Trimester")
{
 returnValue = 3.0;
}
else if (programTermType == "Quarter")
{
 returnValue = 3.0;
}
return returnValue;
```
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Document Management Scripts

These are the prompts for scripts controlling document requirements, acceptance, and ownership.

## Document Owner Logic

Name: documentOwnerLogic

Purpose: Determine the document owners.

In Baseline Config: No

Return Type: Void

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
```

```
## Script Purpose and Context
The script being analyzed is: "documentOwnerLogic"
Primary Function: Determine the document owners.
Expected Output Type: Void
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
allAwardYearReceivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
currentDocument
Purpose: The document being evaluated.
Type: IReceivedDocDocumentAPI
Available Methods:
* getComments
 - Returns: String
 - Function: No description available'
* getDocumentCode
 - Returns: String
 - Function: No description available'
* getDocumentDefn
 - Returns: IDocumentDefnAPI
 - Function: No description available'
* getDocumentFieldMetadataList
 - Returns: List<IDocumentFieldMetadataAPI>
 - Function: No description available'
* getDocumentFieldValue
 - Returns: String
 - Function: No description available'
* getDocumentOwner
 - Returns: String
 - Function: No description available'
* getDocumentOwnerTypeAsString
 - Returns: String
 - Function: No description available'
* getDocumentOwnerTypes
 - Returns: List<String>
 - Function: No description available'
* getDocumentSource
 - Returns: String
 - Function: No description available'
* getDocumentStatus
 - Returns: String
 - Function: No description available'
* getElectronicallyReceived
 - Returns: Boolean
 - Function: No description available'
```

```
* getExternalDocumentId
 - Returns: String
 - Function: No description available'
* getExternalStudentId
 - Returns: String
 - Function: No description available'
* getInZ
 - Returns: Date
 - Function: No description available'
* getOriginalDmsStatus
 - Returns: String
 - Function: No description available'
* getReceiptD
 - Returns: Date
 - Function: No description available'
* getResourceLink
 - Returns: String
 - Function: No description available'
* getReviewD
 - Returns: Date
 - Function: No description available'
* getReviewer
 - Returns: String
 - Function: No description available'
* getSubTypeCode
 - Returns: String
 - Function: No description available'
* getSubTypeIndicator
 - Returns: String
 - Function: No description available'
* isAcceptable
 - Returns: Boolean
 - Function: No description available'
* getAwardYear
 - Returns: String
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
```

```
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isirRecord
Purpose: Current ISIR for the student (the one with the highest CPS number).
Type: IIsirRecordDocumentAPI
Available Methods:
* getActiveIsirForAwardYear
 - Returns: Boolean
 - Function: No description available'
* getAllFields
 - Returns: @NonNull Map<IsirField<?>,IIsirFieldValue<?>>
 - Function: Retrieves all field values (WARNING: can only be used with 2025+ ISIRs).'
* getAwardYear
 - Returns: String
 - Function: No description available'
* getCpsTransNumber
 - Returns: String
 - Function: No description available'
* getCurrentSsn
 - Returns: ISocialSecurityNumber
 - Function: No description available'
* getField
 - Returns: <T> @NonNull IIsirFieldValue<T>
 - Function: Retrieves the value of a given field (WARNING: can only be used with 2025+ ISIRs).'
* getFirstName
 - Returns: String
 - Function: No description available'
* getIsirFieldValue
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValues
```

```
 - Returns: Map<String,String>
 - Function: Get a map containing all ISIR fields (WARNING: This method can only be used up to 2024 award
 year).'
* getIsirIdOriginalNameId
 - Returns: String
 - Function: No description available'
* getIsirIdSsn
 - Returns: ISocialSecurityNumber
 - Function: No description available'
* getIsirIdYearIndicator
 - Returns: String
 - Function: No description available'
* getIsirMatchStatusId
 - Returns: Long
 - Function: No description available'
* getIsirRecordStatusId
 - Returns: Long
 - Function: No description available'
* getIsirStatus
 - Returns: String
 - Function: No description available'
* getIsirStatusBeforeCorrection
 - Returns: String
 - Function: When ISIR Status is Corrected returns the previous ISIR Status.'
* getLastName
 - Returns: String
 - Function: No description available'
* getMiddleName
 - Returns: String
 - Function: No description available'
* getReceivedDate
 - Returns: Date
 - Function: No description available'
* getSchoolCode1
 - Returns: String
 - Function: No description available'
* getSchoolCode10
 - Returns: String
 - Function: No description available'
* getSchoolCode2
 - Returns: String
 - Function: No description available'
* getSchoolCode3
 - Returns: String
 - Function: No description available'
* getSchoolCode4
 - Returns: String
 - Function: No description available'
* getSchoolCode5
 - Returns: String
 - Function: No description available'
* getSchoolCode6
 - Returns: String
 - Function: No description available'
* getSchoolCode7
 - Returns: String
 - Function: No description available'
* getSchoolCode8
 - Returns: String
 - Function: No description available'
* getSchoolCode9
 - Returns: String
 - Function: No description available'
* isCompatibleWithPackaging
 - Returns: boolean
 - Function: No description available'
```

```
* isUnverified
 - Returns: boolean
 - Function: No description available'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
receivedDocuments
Purpose: All documents received for the student being evaluated that match the award year of
 currentDocument.
Type: IReceivedDocs
Available Methods:
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and Student, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and owners, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code, owners, and subtype, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and Student, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and owners, if it exists.'
* getValidDocCodes
 - Returns: Set<String>
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: Return true if doc exists with the right code and owners'
* noDocs
 - Returns: boolean
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
returnList
Purpose: List with a single element that must be populated by the script with a string that determine the
 document
```

```
owner. Accepted values are Student, Mother, Father, Spouse, Parent1 and Parent2. Additional owners
that match one of the accepted values can also be added to the list.
Type: List
Available Methods:
No methods available
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
```

```
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
```

```
    // - [specific edge cases handled]
    // - [how each is managed]
    // Error Handling: [boundary violation management]
    // Impact: [effect of boundary violations]
    // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
   3. ARITHMETIC Section:
    Format:
    // ARITHMETIC:
    // Operation [number]: [name of calculation]
    // Formula: [complete mathematical expression]
    // Variables:
    // - [each variable and its purpose]
    // - [data types and units]
    // Steps:
    // 1. [detailed step-by-step breakdown]
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
   ADDITIONAL REQUIREMENTS:
   1. Complex Logic Sections:
    - Must break down multi-step operations
    - Explain interactions between components
    - Detail data flow and transformations
   2. Error Handling:
    - Document all error conditions
    - Explain recovery procedures
    - Detail logging and reporting
   3. API Interactions:
    - Document all platform method calls
    - Explain data transformations
    - Detail error handling
   4. Financial Aid Impact:
    - Explain effects on aid calculations
    - Document regulatory compliance
    - Detail student eligibility impact
   Example Section With All Aspects:
   // CONDITIONALS:
   // Check 1: Payment Period Status Validation
   // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
   // Purpose: Ensure only valid, active payment periods are included
   // Logic:
   // - Retrieve all loan payment periods
   // - Filter based on status (!= "CANCELED")
   // - Compare start dates with reference date
   // Success Path: Period included in calculations
   // Failure Path: Period excluded from totals
   // Business Rules: Canceled periods must not affect grade level
   // Error Handling: Invalid periods logged and skipped
   // Financial Impact: Affects available loan amounts
   //
   // BOUNDARY CHECKS:
   // Check 1: Payment Period Date Validation
   // Source: Academic Calendar Configuration
   // Variables: period.getStartDate(), referenceDate
   // Validation:
   // - Compare period start date with reference date
   // - Ensure dates are not null
   // Edge Cases:
   // - Null dates are excluded
   // - Future dates are filtered
   // Error Handling: Invalid dates logged and skipped
   // Impact: Prevents future periods from affecting current calculation
```

```
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
N/A - There is no baseline config. This is a placeholder.
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Document FA Acceptance

Name: documentFaAcceptance

Purpose: Determine the financial aid acceptance of a document.

In Baseline Config: No

Return Type: Boolean

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "documentFaAcceptance"
Primary Function: Determine the Financial Aid (FA) acceptance of a document. This script is triggered by
 calls to
IReceivedDoc.isAcceptable() and IReceivedDocDocumentAPI.isAcceptable() originated from other Groovy scripts.
Expected Output Type: Boolean
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
```

```
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
currentDocument
Purpose: Input document being evaluated, representing the API object from which the call to isAcceptable()
 is
originating.
Type: IReceivedDoc
Available Methods:
* getComments
 - Returns: String
 - Function: No description available'
* getDocumentDefn
 - Returns: IDocumentDefnAPI
 - Function: No description available'
* getDocumentField
 - Returns: String
 - Function: No description available'
* getDocumentFieldMetadataList
 - Returns: List<IDocumentFieldMetadataAPI>
 - Function: No description available'
* getDocumentFieldValuesMap
 - Returns: Map<String,String>
 - Function: No description available'
* getDocumentMetadata
 - Returns: Map<String,String>
 - Function: No description available'
* getAwardYear
 - Returns: String
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getDocumentOwner
 - Returns: String
 - Function: No description available'
* getDocumentOwnerTypes
 - Returns: List<String>
 - Function: No description available'
* getDocumentSource
 - Returns: String
 - Function: No description available'
* getElectronicallyReceived
 - Returns: Boolean
 - Function: No description available'
* getExternalDocumentId
 - Returns: String
 - Function: No description available'
* getExternalStudentId
 - Returns: String
 - Function: No description available'
* getInZ
 - Returns: Date
 - Function: No description available'
* getOriginalDmsStatus
 - Returns: String
 - Function: No description available'
```

```
* getReceiptDate
 - Returns: Date
 - Function: No description available'
* getResourceLink
 - Returns: String
 - Function: No description available'
* getReviewDate
 - Returns: Date
 - Function: No description available'
* getReviewer
 - Returns: String
 - Function: No description available'
* getSubTypeCode
 - Returns: String
 - Function: No description available'
* getSubTypeValue
 - Returns: String
 - Function: No description available'
* isAcceptable
 - Returns: Boolean
 - Function: No description available'
isirRecord
Purpose: The same isirRecord binding from the script that called method isAcceptable().
Type: IIsirRecordAPI
Available Methods:
* getActiveIsirForAwardYear
 - Returns: Boolean
 - Function: Gets the active ISIR for award year status.'
* getAllFields
 - Returns: @NonNull Map<IsirField<?>,IIsirFieldValue<?>>
 - Function: Retrieves all field values (WARNING: can only be used with 2025+ ISIRs).'
* getAssumptions
 - Returns: List<IAssumptionAPI>
 - Function: Get the assumptions.'
* getAwardYear
 - Returns: String
 - Function: Gets the award year.'
* getCommentCodes
 - Returns: List<ICommentCodeAPI>
 - Function: Get the comment codes.'
* getCpsTransNumber
 - Returns: String
 - Function: Gets the CPS transaction number.'
* getCurrentSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's current SSN.'
* getDiscrepancies
 - Returns: List<IDiscrepancyAPI>
 - Function: Get the discrepancies.'
* getField
 - Returns: <T> @NonNull IIsirFieldValue<T>
 - Function: Retrieves the value of a given field (WARNING: can only be used with 2025+ ISIRs).'
* getFirstName
 - Returns: String
 - Function: Gets the student's first name.'
* getHighlightFlags
 - Returns: List<IHighlightFlagAPI>
 - Function: Get the highlight flags.'
* getIsirFieldValue
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValueOrNull
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValues
 - Returns: Map<String,String>
```

```
 - Function: Get a map containing all ISIR fields (WARNING: This method can only be used up to 2024 award
 year).'
* getIsirIdOriginalNameId
 - Returns: String
 - Function: Gets the student's original name ID.'
* getIsirIdSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's original SSN.'
* getIsirIdYearIndicator
 - Returns: String
 - Function: Gets the ISIR's year indicator.'
* getIsirStatus
 - Returns: String
 - Function: Gets the ISIR status.'
* getIsirStatusBeforeCorrection
 - Returns: String
 - Function: When ISIR Status is Corrected returns the previous ISIR Status.'
* getLastName
 - Returns: String
 - Function: Gets the student's last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the student's middle name.'
* getReceivedDate
 - Returns: Date
 - Function: Gets the system receipt date of the ISIR.'
* getRejectCodes
 - Returns: List<IRejectCodeAPI>
 - Function: Get the reject codes.'
* getSchoolCode1
 - Returns: String
 - Function: Gets the school code 1.'
* getSchoolCode10
 - Returns: String
 - Function: Gets the school code 10.'
* getSchoolCode2
 - Returns: String
 - Function: Gets the school code 2.'
* getSchoolCode3
 - Returns: String
 - Function: Gets the school code 3.'
* getSchoolCode4
 - Returns: String
 - Function: Gets the school code 4.'
* getSchoolCode5
 - Returns: String
 - Function: Gets the school code 5.'
* getSchoolCode6
 - Returns: String
 - Function: Gets the school code 6.'
* getSchoolCode7
 - Returns: String
 - Function: Gets the school code 7.'
* getSchoolCode8
 - Returns: String
 - Function: Gets the school code 8.'
* getSchoolCode9
 - Returns: String
 - Function: Gets the school code 9.'
* isCompatibleWithPackaging
 - Returns: boolean
 - Function: Checks if this ISIR satisfies minimal acceptance criteria for automatic Financial
Aid Package Calculation.'
* isUnverified
 - Returns: boolean
 - Function: Gets the ISIR verification status.'
```

```
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
receivedDocuments
Purpose: The same receivedDocuments binding from the script that called method isAcceptable().
Type: IReceivedDocs
Available Methods:
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and Student, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and owners, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code, owners, and subtype, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and Student, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and owners, if it exists.'
* getValidDocCodes
 - Returns: Set<String>
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: Return true if doc exists with the right code and owners'
* noDocs
 - Returns: boolean
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
student
Purpose: The same student binding from the script that called method isAcceptable().
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
```

```
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
```

```
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
 // Operation [number]: [name of calculation]
 // Formula: [complete mathematical expression]
 // Variables:
 // - [each variable and its purpose]
```

```
 // - [data types and units]
 // Steps:
 // 1. [detailed step-by-step breakdown]
 // 2. [intermediate calculations]
 // 3. [transformations applied]
 // Business Context: [why this calculation is needed]
 // Result: [meaning of the output]
 // Financial Impact: [effect on aid calculations]
 // Limitations: [any mathematical constraints]
 // If N/A, explicitly state: "ARITHMETIC: N/A"
ADDITIONAL REQUIREMENTS:
1. Complex Logic Sections:
 - Must break down multi-step operations
 - Explain interactions between components
 - Detail data flow and transformations
2. Error Handling:
 - Document all error conditions
 - Explain recovery procedures
 - Detail logging and reporting
3. API Interactions:
 - Document all platform method calls
 - Explain data transformations
 - Detail error handling
4. Financial Aid Impact:
 - Explain effects on aid calculations
 - Document regulatory compliance
 - Detail student eligibility impact
Example Section With All Aspects:
// CONDITIONALS:
// Check 1: Payment Period Status Validation
// Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
```

```
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
N/A - There is no baseline config. This is a placeholder.
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Document Cancellation

Name: documentCancellation

Purpose: Determine when a document request can be canceled.

In Baseline Config: No

Return Type: Void

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "documentCancellation"
Primary Function: Determine when a document request can be canceled.
Expected Output Type: Void
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
allAwardYearReceivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
```

```
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
currentDocument
Purpose: The document being evaluated.
Type: IReceivedDocDocumentAPI
Available Methods:
* getComments
 - Returns: String
 - Function: No description available'
* getDocumentCode
 - Returns: String
 - Function: No description available'
* getDocumentDefn
 - Returns: IDocumentDefnAPI
 - Function: No description available'
* getDocumentFieldMetadataList
 - Returns: List<IDocumentFieldMetadataAPI>
 - Function: No description available'
* getDocumentFieldValue
 - Returns: String
 - Function: No description available'
* getDocumentOwner
 - Returns: String
 - Function: No description available'
* getDocumentOwnerTypeAsString
 - Returns: String
 - Function: No description available'
* getDocumentOwnerTypes
 - Returns: List<String>
 - Function: No description available'
* getDocumentSource
 - Returns: String
 - Function: No description available'
* getDocumentStatus
 - Returns: String
 - Function: No description available'
* getElectronicallyReceived
 - Returns: Boolean
 - Function: No description available'
* getExternalDocumentId
 - Returns: String
 - Function: No description available'
* getExternalStudentId
 - Returns: String
 - Function: No description available'
* getInZ
 - Returns: Date
 - Function: No description available'
* getOriginalDmsStatus
 - Returns: String
 - Function: No description available'
* getReceiptD
 - Returns: Date
 - Function: No description available'
* getResourceLink
 - Returns: String
 - Function: No description available'
* getReviewD
 - Returns: Date
 - Function: No description available'
* getReviewer
 - Returns: String
```

```
 - Function: No description available'
* getSubTypeCode
 - Returns: String
 - Function: No description available'
* getSubTypeIndicator
 - Returns: String
 - Function: No description available'
* isAcceptable
 - Returns: Boolean
 - Function: No description available'
* getAwardYear
 - Returns: String
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
```

```
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isirRecord
Purpose: Current ISIR for the student (the one with the highest CPS number).
Type: IIsirRecordDocumentAPI
Available Methods:
* getActiveIsirForAwardYear
 - Returns: Boolean
 - Function: No description available'
* getAllFields
 - Returns: @NonNull Map<IsirField<?>,IIsirFieldValue<?>>
 - Function: Retrieves all field values (WARNING: can only be used with 2025+ ISIRs).'
* getAwardYear
 - Returns: String
 - Function: No description available'
* getCpsTransNumber
 - Returns: String
 - Function: No description available'
* getCurrentSsn
 - Returns: ISocialSecurityNumber
 - Function: No description available'
* getField
 - Returns: <T> @NonNull IIsirFieldValue<T>
 - Function: Retrieves the value of a given field (WARNING: can only be used with 2025+ ISIRs).'
* getFirstName
 - Returns: String
 - Function: No description available'
* getIsirFieldValue
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValues
 - Returns: Map<String,String>
 - Function: Get a map containing all ISIR fields (WARNING: This method can only be used up to 2024 award
 year).'
* getIsirIdOriginalNameId
 - Returns: String
 - Function: No description available'
* getIsirIdSsn
 - Returns: ISocialSecurityNumber
 - Function: No description available'
* getIsirIdYearIndicator
 - Returns: String
 - Function: No description available'
* getIsirMatchStatusId
 - Returns: Long
 - Function: No description available'
* getIsirRecordStatusId
 - Returns: Long
 - Function: No description available'
* getIsirStatus
 - Returns: String
 - Function: No description available'
* getIsirStatusBeforeCorrection
 - Returns: String
```

```
 - Function: When ISIR Status is Corrected returns the previous ISIR Status.'
* getLastName
 - Returns: String
 - Function: No description available'
* getMiddleName
 - Returns: String
 - Function: No description available'
* getReceivedDate
 - Returns: Date
 - Function: No description available'
* getSchoolCode1
 - Returns: String
 - Function: No description available'
* getSchoolCode10
 - Returns: String
 - Function: No description available'
* getSchoolCode2
 - Returns: String
 - Function: No description available'
* getSchoolCode3
 - Returns: String
 - Function: No description available'
* getSchoolCode4
 - Returns: String
 - Function: No description available'
* getSchoolCode5
 - Returns: String
 - Function: No description available'
* getSchoolCode6
 - Returns: String
 - Function: No description available'
* getSchoolCode7
 - Returns: String
 - Function: No description available'
* getSchoolCode8
 - Returns: String
 - Function: No description available'
* getSchoolCode9
 - Returns: String
 - Function: No description available'
* isCompatibleWithPackaging
 - Returns: boolean
 - Function: No description available'
* isUnverified
 - Returns: boolean
 - Function: No description available'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
```

```
 - Returns: void
 - Function: No description available'
receivedDocuments
Purpose: All documents received for the student being evaluated that match the award year of
 currentDocument.
Type: IReceivedDocs
Available Methods:
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and Student, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and owners, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code, owners, and subtype, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and Student, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and owners, if it exists.'
* getValidDocCodes
 - Returns: Set<String>
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: Return true if doc exists with the right code and owners'
* noDocs
 - Returns: boolean
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
requestedDocIdToCancel
Purpose: List to be populated by the script with document IDs to be cancelled.
Type: List
Available Methods:
No methods available
requestedDocs
Purpose: Open document requests for the student. If currentDocument has an award year, it includes only
 requests for
that award year. Otherwise, it includes all open requests for the student.
Type: IDocumentRequestsAPI
Available Methods:
* get
 - Returns: IDocumentRequestAPI
 - Function: Return IDocumentRequestAPI for specified code and owner, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IDocumentRequestAPI>
 - Function: Return list with IDocumentRequestAPIs for specified code and Student, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IDocumentRequestAPI>
 - Function: Return list with IDocumentRequestAPI for specified code and owner, if it exists.'
* getValidDocCodes
 - Returns: Set<String>
 - Function: No description available'
* hasReqDoc
 - Returns: boolean
 - Function: No description available'
```

```
* hasReqDoc
 - Returns: boolean
 - Function: Return true if IDocumentRequestAPI exists with the right code and owner'
* noDocs
 - Returns: boolean
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
* get
 - Returns: IDocumentRequestAPI
 - Function: Return the IDocumentRequestAPI for specified code and Student, if it exists.'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
```

```
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
```

```
    // Check [number]: [specific limit being enforced]
    // Source: [where boundary values come from]
    // Variables: [variables and methods involved]
    // Validation:
    // - [how boundaries are checked]
    // - [validation method details]
    // Edge Cases:
    // - [specific edge cases handled]
    // - [how each is managed]
    // Error Handling: [boundary violation management]
    // Impact: [effect of boundary violations]
    // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
  3. ARITHMETIC Section:
    Format:
    // ARITHMETIC:
    // Operation [number]: [name of calculation]
    // Formula: [complete mathematical expression]
    // Variables:
    // - [each variable and its purpose]
    // - [data types and units]
    // Steps:
    // 1. [detailed step-by-step breakdown]
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
  ADDITIONAL REQUIREMENTS:
  1. Complex Logic Sections:
    - Must break down multi-step operations
    - Explain interactions between components
    - Detail data flow and transformations
  2. Error Handling:
    - Document all error conditions
    - Explain recovery procedures
    - Detail logging and reporting
  3. API Interactions:
    - Document all platform method calls
    - Explain data transformations
    - Detail error handling
  4. Financial Aid Impact:
    - Explain effects on aid calculations
    - Document regulatory compliance
    - Detail student eligibility impact
  Example Section With All Aspects:
  // CONDITIONALS:
  // Check 1: Payment Period Status Validation
  // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
  // Purpose: Ensure only valid, active payment periods are included
  // Logic:
  // - Retrieve all loan payment periods
  // - Filter based on status (!= "CANCELED")
  // - Compare start dates with reference date
  // Success Path: Period included in calculations
  // Failure Path: Period excluded from totals
  // Business Rules: Canceled periods must not affect grade level
  // Error Handling: Invalid periods logged and skipped
  // Financial Impact: Affects available loan amounts
  //
  // BOUNDARY CHECKS:
  // Check 1: Payment Period Date Validation
  // Source: Academic Calendar Configuration
  // Variables: period.getStartDate(), referenceDate
  // Validation:
```

```
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
N/A - There is no baseline config. This is a placeholder.
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Document Metadata Additional Resolution Action

Name: documentMetadataAdditionalResolutionAction

Purpose: Evaluate document metadata fields for document acceptance.

In Baseline Config: Yes

Return Type: Void

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "documentMetadataAdditionalResolutionAction"
Primary Function: Evaluate document metadata fields for document acceptance and required supporting
 documents.
```

```
Expected Output Type: Void
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
allAwardYearReceivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
currentDocument
Purpose: The document being evaluated.
Type: IReceivedDocDocumentAPI
Available Methods:
* getComments
 - Returns: String
 - Function: No description available'
* getDocumentCode
 - Returns: String
 - Function: No description available'
* getDocumentDefn
 - Returns: IDocumentDefnAPI
 - Function: No description available'
* getDocumentFieldMetadataList
 - Returns: List<IDocumentFieldMetadataAPI>
 - Function: No description available'
* getDocumentFieldValue
 - Returns: String
 - Function: No description available'
* getDocumentOwner
 - Returns: String
 - Function: No description available'
* getDocumentOwnerTypeAsString
 - Returns: String
 - Function: No description available'
* getDocumentOwnerTypes
 - Returns: List<String>
 - Function: No description available'
* getDocumentSource
 - Returns: String
 - Function: No description available'
* getDocumentStatus
 - Returns: String
 - Function: No description available'
* getElectronicallyReceived
 - Returns: Boolean
 - Function: No description available'
* getExternalDocumentId
 - Returns: String
 - Function: No description available'
```

```
* getExternalStudentId
 - Returns: String
 - Function: No description available'
* getInZ
 - Returns: Date
 - Function: No description available'
* getOriginalDmsStatus
 - Returns: String
 - Function: No description available'
* getReceiptD
 - Returns: Date
 - Function: No description available'
* getResourceLink
 - Returns: String
 - Function: No description available'
* getReviewD
 - Returns: Date
 - Function: No description available'
* getReviewer
 - Returns: String
 - Function: No description available'
* getSubTypeCode
 - Returns: String
 - Function: No description available'
* getSubTypeIndicator
 - Returns: String
 - Function: No description available'
* isAcceptable
 - Returns: Boolean
 - Function: No description available'
* getAwardYear
 - Returns: String
 - Function: No description available'
docMetadataFieldIndex
Purpose: Index of the metadata field being evaluated, starting from zero.
Type: Integer
Available Methods:
No methods available
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
```

```
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isirRecord
Purpose: Current ISIR for the student (the one with the highest CPS number).
Type: IIsirRecordDocumentAPI
Available Methods:
* getActiveIsirForAwardYear
 - Returns: Boolean
 - Function: No description available'
* getAllFields
 - Returns: @NonNull Map<IsirField<?>,IIsirFieldValue<?>>
 - Function: Retrieves all field values (WARNING: can only be used with 2025+ ISIRs).'
* getAwardYear
 - Returns: String
 - Function: No description available'
* getCpsTransNumber
 - Returns: String
 - Function: No description available'
* getCurrentSsn
 - Returns: ISocialSecurityNumber
 - Function: No description available'
* getField
 - Returns: <T> @NonNull IIsirFieldValue<T>
 - Function: Retrieves the value of a given field (WARNING: can only be used with 2025+ ISIRs).'
* getFirstName
 - Returns: String
 - Function: No description available'
* getIsirFieldValue
 - Returns: String
```

```
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValues
 - Returns: Map<String,String>
 - Function: Get a map containing all ISIR fields (WARNING: This method can only be used up to 2024 award
 year).'
* getIsirIdOriginalNameId
 - Returns: String
 - Function: No description available'
* getIsirIdSsn
 - Returns: ISocialSecurityNumber
 - Function: No description available'
* getIsirIdYearIndicator
 - Returns: String
 - Function: No description available'
* getIsirMatchStatusId
 - Returns: Long
 - Function: No description available'
* getIsirRecordStatusId
 - Returns: Long
 - Function: No description available'
* getIsirStatus
 - Returns: String
 - Function: No description available'
* getIsirStatusBeforeCorrection
 - Returns: String
 - Function: When ISIR Status is Corrected returns the previous ISIR Status.'
* getLastName
 - Returns: String
 - Function: No description available'
* getMiddleName
 - Returns: String
 - Function: No description available'
* getReceivedDate
 - Returns: Date
 - Function: No description available'
* getSchoolCode1
 - Returns: String
 - Function: No description available'
* getSchoolCode10
 - Returns: String
 - Function: No description available'
* getSchoolCode2
 - Returns: String
 - Function: No description available'
* getSchoolCode3
 - Returns: String
 - Function: No description available'
* getSchoolCode4
 - Returns: String
 - Function: No description available'
* getSchoolCode5
 - Returns: String
 - Function: No description available'
* getSchoolCode6
 - Returns: String
 - Function: No description available'
* getSchoolCode7
 - Returns: String
 - Function: No description available'
* getSchoolCode8
 - Returns: String
 - Function: No description available'
* getSchoolCode9
 - Returns: String
 - Function: No description available'
* isCompatibleWithPackaging
```

```
 - Returns: boolean
 - Function: No description available'
* isUnverified
 - Returns: boolean
 - Function: No description available'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
receivedDocuments
Purpose: All documents received for the student being evaluated that match the award year of
 currentDocument.
Type: IReceivedDocs
Available Methods:
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and Student, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and owners, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code, owners, and subtype, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and Student, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and owners, if it exists.'
* getValidDocCodes
 - Returns: Set<String>
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: Return true if doc exists with the right code and owners'
* noDocs
 - Returns: boolean
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
returnList
```

```
Purpose: List with a single element, for which method setScriptReturnErrorString() must be called with an
 error
string as parameter in case the document is not accepted.
Type: List
Available Methods:
No methods available
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
```

```
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
supportingDoc
Purpose: Binding to be populated with additional supporting documents required based on the evaluation of
 the current
field.
Type: IIsirRequiredDocsAPI
Available Methods:
* addAnyDocuments
 - Returns: void
 - Function: Adds a list of documents, any one in the list is required.'
* addAnyDocuments
 - Returns: void
 - Function: Adds a list of documents, any one in the list is required.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* any
 - Returns: IIsirRequiredDocsAPI
 - Function: No description available'
* clear
 - Returns: void
 - Function: Clears all docs.'
* doc
 - Returns: IIsirRequiredDocsAPI
 - Function: No description available'
* getAdditionalDocRequestMessages
 - Returns: List<String>
 - Function: No description available'
* getValidDocCodes
 - Returns: Set<String>
 - Function: No description available'
* hasDocument
 - Returns: boolean
 - Function: Returns true if the document has been added, either as a regular request
or an 'any' request.'
* hasDocument
 - Returns: boolean
 - Function: Returns true if the document has been added, either as a regular request
or an 'any' request.'
* isEmpty
 - Returns: boolean
 - Function: Checks if these requirements are empty.'
```

```
* setValidDocCodes
 - Returns: void
 - Function: No description available'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
 // Operation [number]: [name of calculation]
 // Formula: [complete mathematical expression]
 // Variables:
 // - [each variable and its purpose]
 // - [data types and units]
 // Steps:
 // 1. [detailed step-by-step breakdown]
```

```
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
ADDITIONAL REQUIREMENTS:
1. Complex Logic Sections:
 - Must break down multi-step operations
 - Explain interactions between components
 - Detail data flow and transformations
2. Error Handling:
 - Document all error conditions
 - Explain recovery procedures
 - Detail logging and reporting
3. API Interactions:
 - Document all platform method calls
 - Explain data transformations
 - Detail error handling
4. Financial Aid Impact:
 - Explain effects on aid calculations
 - Document regulatory compliance
 - Detail student eligibility impact
Example Section With All Aspects:
// CONDITIONALS:
// Check 1: Payment Period Status Validation
// Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
```

```
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.DocumentMetadataAdditionalResolutionActio
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
//Variable definitions
def docOwner = currentDocument.getDocumentOwnerTypes().get(0)
def filingStatus = currentDocument.getDocumentFieldValue("AC1118")
if (docOwner == "Student" && filingStatus != null && !filingStatus.isAllWhitespace() && filingStatus ==
 "Married-Filed Separate Return")
{
 supportingDoc.addAnyDocuments(["TaxReturnTranscript", "Student Spouse"], ["1040", "Student Spouse"],
 ["1040x", "Student Spouse"],["ForeignTaxTranscript", "Student Spouse"])
}
/*if (docOwner == "Parent" && filingStatus != null && !filingStatus.isAllWhitespace() && filingStatus ==
 "Married Filing Separately")
{
 supportingDoc.addAnyDocuments(["TaxReturnTranscript", "Parent Spouse"], ["1040", "Parent Spouse"],
 ["1040x", "Parent Spouse"],["ForeignTaxTranscript", "Parent Spouse"])
}*/
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Nonfederal (NFR) Resources Scripts

These are the prompts for scripts managing nonfederal resources, including eligibility, periods, and award limits.

## NFR Eligibility

Name: nfrEligibility

Purpose: Evaluates eligibility for an NFR fund.

In Baseline Config: Yes

Return Type: List<NfrEligibilityField>

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
```

```
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "nfrEligibility"
Primary Function: Evaluates eligibility for an NFR fund.
Expected Output Type: List<NfrEligibilityField>
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
acy
Purpose: The academic year of the funding element being evaluated.
Type: IAcademicYearScheduleAPI
Available Methods:
* getCompletedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getPeriodOfEnrollmentEndDate
 - Returns: LocalDate
 - Function: No description available'
* getPeriodOfEnrollmentStartDate
 - Returns: LocalDate
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
acyNo
Purpose: The number of the academic year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
acyOrder
```

```
Purpose: The order number of the academic year of the funding element being evaluated, among all non-
canceled academic
years.
Type: Integer
Available Methods:
No methods available
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
```

```
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
```

```
fundCode
Purpose: The fund code of the funding element being evaluated.
Type: String
Available Methods:
No methods available
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
fundType
Purpose: The fund type of the funding element being evaluated.
Type: String
Available Methods:
No methods available
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isFirstAcy
Purpose: True if the academic year of the funding element being evaluated is the first one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isLastAcy
```

```
Purpose: True if the academic year of the funding element being evaluated is the last one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
nfrConfig
Purpose: Interface that allows calling other NFR scripts for the same fund and student.
Type: INfrConfigAPI
Available Methods:
* evalAggregateLimit
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalAggregateLimit script for the same fund configuration and
 student as used
to evaluate the calling script.'
* evalAnnualLimit
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalAnnualLimit for the same fund configuration and student as used
 to
evaluate the calling script.'
* evalAwardingCriteria
 - Returns: Boolean
 - Function: Evaluate awardability of NFR fund for a given period number'
* evalDisbursementDate
 - Returns: LocalDate
 - Function: Equivalent to a call to the evalDisbursementDates script for the same fund configuration and
 student as
used to evaluate the calling script.'
* evalEligibility
 - Returns: INfrEligibilityResult
 - Function: Equivalent to a call to the evalEligibility script for the same fund configuration and student
 as used to
```

```
evaluate the calling script.'
* evalMaximumProjectedAwardAmount
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalMaximumProjectedAwardAmount script for the same fund
 configuration and
student as used to evaluate the calling script.'
* evalMinimumAmount
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalMinimumAmount script for the same fund configuration and
 student as used
to evaluate the calling script.'
* evalPeriods
 - Returns: IPeriods<NfrPeriod,?>
 - Function: Equivalent to a call to the evalPeriods script for the same fund configuration and student as
 used to evaluate the calling script.'
* getAwardYear
 - Returns: int
 - Function: Gets the award year of the configuration.'
* getAwardYearLimit
 - Returns: BigDecimal
 - Function: Gets the award year limit for the fund.'
nfrConfigs
Purpose: Maps (end years of) award years to the respective configuration in the current academic year
 calculation. Allows
calling other NFR scripts for the same fund and student but for different active award years in this
 academic year.
The user can rely on the entries in this map being ordered by award year.
Type: Map
Available Methods:
No methods available
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
```

```
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
```

```
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
```

```
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
```

```
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
Available Methods:
* getPeriodsIntersectingWithGroupForThePeriod
```

```
 - Returns: List<? extends IPeriod>
 - Function: Find the periods student was in the given group in the specified period of time'
* isInGroupAtAnyPointInThePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period at any point of time.'
* isInGroupAtDate
 - Returns: boolean
 - Function: Checks if student was in the given group on the given date.'
* isInGroupDuringCompletePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period for the whole time.'
* numberOfDaysInGroupForThePeriod
 - Returns: int
 - Function: Count number of days student was in the given group in the specified period of time'
* getActiveGroupsAtAnyPointInThePeriod
 - Returns: List<String>
 - Function: Find the student's groups active at any point during specified period of time'
* getActiveGroupsAtDate
 - Returns: List<String>
 - Function: Find the student's active groups at the given date'
* getActiveGroupsDuringCompletePeriod
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
acys
Purpose: Provides access to all non-canceled academic years.
Type: IAcademicYearsScheduleAPI
Available Methods:
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
awy
Purpose: The award year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
awyPeriod
Purpose: The period of the award year of the funding element being evaluated.
Type: IAwardYearAPI
Available Methods:
* asList
 - Returns: IAwardYearsAPI
 - Function: Returns an IAwardYearsAPI instance containing only this Award Year.'
* getEndYear
 - Returns: int
 - Function: Gets the end year of this Award Year, for example 2023 for Award Year 2022-2023.'
* getNext
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that follows this one chronologically.'
* getPrevious
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that precedes this one chronologically.'
* getStartYear
 - Returns: int
 - Function: Gets the start year of this Award Year, for example 2022 for Award Year 2022-2023.'
* getYear
```

```
 - Returns: int
 - Function: Short for getEndYear().'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
No methods available
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
```

```
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
```

```
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
```

```
    // Success Path: [what happens when condition is true]
    // Failure Path: [what happens when condition is false]
    // Business Rules: [specific rules being enforced]
    // Error Handling: [how errors are managed]
    // Financial Impact: [effect on aid calculations]
    // If N/A, explicitly state: "CONDITIONALS: N/A"
   2. BOUNDARY CHECKS Section:
    Format:
    // BOUNDARY CHECKS:
    // Check [number]: [specific limit being enforced]
    // Source: [where boundary values come from]
    // Variables: [variables and methods involved]
    // Validation:
    // - [how boundaries are checked]
    // - [validation method details]
    // Edge Cases:
    // - [specific edge cases handled]
    // - [how each is managed]
    // Error Handling: [boundary violation management]
    // Impact: [effect of boundary violations]
    // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
   3. ARITHMETIC Section:
    Format:
    // ARITHMETIC:
    // Operation [number]: [name of calculation]
    // Formula: [complete mathematical expression]
    // Variables:
    // - [each variable and its purpose]
    // - [data types and units]
    // Steps:
    // 1. [detailed step-by-step breakdown]
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
   ADDITIONAL REQUIREMENTS:
   1. Complex Logic Sections:
    - Must break down multi-step operations
    - Explain interactions between components
    - Detail data flow and transformations
   2. Error Handling:
    - Document all error conditions
    - Explain recovery procedures
    - Detail logging and reporting
   3. API Interactions:
    - Document all platform method calls
    - Explain data transformations
    - Detail error handling
   4. Financial Aid Impact:
    - Explain effects on aid calculations
    - Document regulatory compliance
    - Detail student eligibility impact
   Example Section With All Aspects:
   // CONDITIONALS:
   // Check 1: Payment Period Status Validation
   // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
   // Purpose: Ensure only valid, active payment periods are included
   // Logic:
   // - Retrieve all loan payment periods
   // - Filter based on status (!= "CANCELED")
   // - Compare start dates with reference date
   // Success Path: Period included in calculations
   // Failure Path: Period excluded from totals
```

```
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.NfrEligibilityScript)
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
def NfrEligibilityField automatedFundEligibility = new NfrEligibilityField("Automated Fund", true)
return [automatedFundEligibility]
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# NFR Periods

Name: nfrPeriods

Purpose: Evaluates periods for an NFR fund.

In Baseline Config: Yes

Return Type: List<NfrPeriod>

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "nfrPeriods"
Primary Function: Evaluates periods for an NFR fund.
Expected Output Type: List<NfrPeriod>
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
acy
Purpose: The academic year of the funding element being evaluated.
Type: IAcademicYearScheduleAPI
Available Methods:
* getCompletedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getPeriodOfEnrollmentEndDate
 - Returns: LocalDate
 - Function: No description available'
* getPeriodOfEnrollmentStartDate
 - Returns: LocalDate
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
```

```
 - Function: No description available'
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
acyNo
Purpose: The number of the academic year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
acyOrder
Purpose: The order number of the academic year of the funding element being evaluated, among all non-
canceled academic
years.
Type: Integer
Available Methods:
No methods available
acys
Purpose: Provides access to all non-canceled academic years.
Type: IAcademicYearsScheduleAPI
Available Methods:
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
awy
Purpose: The award year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
awyPeriod
Purpose: The period of the award year of the funding element being evaluated.
Type: IAwardYearAPI
Available Methods:
* asList
 - Returns: IAwardYearsAPI
 - Function: Returns an IAwardYearsAPI instance containing only this Award Year.'
* getEndYear
 - Returns: int
 - Function: Gets the end year of this Award Year, for example 2023 for Award Year 2022-2023.'
* getNext
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that follows this one chronologically.'
* getPrevious
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that precedes this one chronologically.'
* getStartYear
 - Returns: int
 - Function: Gets the start year of this Award Year, for example 2022 for Award Year 2022-2023.'
* getYear
 - Returns: int
 - Function: Short for getEndYear().'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
```

```
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
No methods available
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
```

```
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
```

```
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
financialPlan
Purpose: Information about the student's current financial plan.
```

```
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
```

```
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCode
Purpose: The fund code of the funding element being evaluated.
Type: String
Available Methods:
No methods available
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
fundType
Purpose: The fund type of the funding element being evaluated.
Type: String
Available Methods:
No methods available
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
```

```
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isFirstAcy
Purpose: True if the academic year of the funding element being evaluated is the first one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isLastAcy
Purpose: True if the academic year of the funding element being evaluated is the last one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
lp
Purpose: The loan period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
```

```
 - Function: No description available'
lpNo
Purpose: The number of the loan period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
nfrConfig
Purpose: Interface that allows calling other NFR scripts for the same fund and student.
Type: INfrConfigAPI
Available Methods:
* evalAggregateLimit
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalAggregateLimit script for the same fund configuration and
 student as used
to evaluate the calling script.'
* evalAnnualLimit
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalAnnualLimit for the same fund configuration and student as used
 to
evaluate the calling script.'
* evalAwardingCriteria
 - Returns: Boolean
 - Function: Evaluate awardability of NFR fund for a given period number'
* evalDisbursementDate
 - Returns: LocalDate
 - Function: Equivalent to a call to the evalDisbursementDates script for the same fund configuration and
 student as
used to evaluate the calling script.'
* evalEligibility
 - Returns: INfrEligibilityResult
 - Function: Equivalent to a call to the evalEligibility script for the same fund configuration and student
 as used to
evaluate the calling script.'
* evalMaximumProjectedAwardAmount
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalMaximumProjectedAwardAmount script for the same fund
 configuration and
student as used to evaluate the calling script.'
* evalMinimumAmount
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalMinimumAmount script for the same fund configuration and
 student as used
to evaluate the calling script.'
* evalPeriods
 - Returns: IPeriods<NfrPeriod,?>
 - Function: Equivalent to a call to the evalPeriods script for the same fund configuration and student as
 used to evaluate the calling script.'
* getAwardYear
 - Returns: int
 - Function: Gets the award year of the configuration.'
* getAwardYearLimit
 - Returns: BigDecimal
 - Function: Gets the award year limit for the fund.'
nfrConfigs
Purpose: Maps (end years of) award years to the respective configuration in the current academic year
 calculation. Allows
calling other NFR scripts for the same fund and student but for different active award years in this
 academic year.
The user can rely on the entries in this map being ordered by award year.
Type: Map
Available Methods:
No methods available
pp
Purpose: The payment period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
```

```
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
ppNo
Purpose: The number of the payment period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
```

```
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
```

```
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
```

```
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
scope
Purpose: Gathers all the scope details of the script being evaluated (acyNo, fundCode,
fundType, awy, lpNo, ppNo) allowing quick access to financialPlan/disbursements/etc
subsets directly pertaining to this scope.
Type: IFundScopeAPI
Available Methods:
* getAcyNo
 - Returns: int
 - Function: No description available'
* getAwy
 - Returns: int
 - Function: No description available'
* getFundCode
 - Returns: String
 - Function: No description available'
* getFundType
 - Returns: String
 - Function: No description available'
* getLpNo
 - Returns: int
 - Function: No description available'
* getPpNo
 - Returns: int
 - Function: No description available'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
```

```
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
```

```
Available Methods:
* getPeriodsIntersectingWithGroupForThePeriod
 - Returns: List<? extends IPeriod>
 - Function: Find the periods student was in the given group in the specified period of time'
* isInGroupAtAnyPointInThePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period at any point of time.'
* isInGroupAtDate
 - Returns: boolean
 - Function: Checks if student was in the given group on the given date.'
* isInGroupDuringCompletePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period for the whole time.'
* numberOfDaysInGroupForThePeriod
 - Returns: int
 - Function: Count number of days student was in the given group in the specified period of time'
* getActiveGroupsAtAnyPointInThePeriod
 - Returns: List<String>
 - Function: Find the student's groups active at any point during specified period of time'
* getActiveGroupsAtDate
 - Returns: List<String>
 - Function: Find the student's active groups at the given date'
* getActiveGroupsDuringCompletePeriod
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
term
Purpose: The term of the funding element being evaluated.
Type: IAcademicTermPeriodAPI
Available Methods:
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getName
 - Returns: String
 - Function: No description available'
* isAttending
 - Returns: boolean
 - Function: No description available'
* isNotAttending
 - Returns: boolean
 - Function: No description available'
* isStandard
 - Returns: boolean
 - Function: No description available'
* isSummer
 - Returns: boolean
 - Function: No description available'
* isSummerHeader
 - Returns: boolean
 - Function: No description available'
* isSummerTrailer
 - Returns: boolean
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
termNo
Purpose: The number of the term of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
```

```
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
 // Operation [number]: [name of calculation]
 // Formula: [complete mathematical expression]
 // Variables:
 // - [each variable and its purpose]
 // - [data types and units]
 // Steps:
 // 1. [detailed step-by-step breakdown]
 // 2. [intermediate calculations]
 // 3. [transformations applied]
 // Business Context: [why this calculation is needed]
 // Result: [meaning of the output]
 // Financial Impact: [effect on aid calculations]
```

```
 // Limitations: [any mathematical constraints]
 // If N/A, explicitly state: "ARITHMETIC: N/A"
ADDITIONAL REQUIREMENTS:
1. Complex Logic Sections:
 - Must break down multi-step operations
 - Explain interactions between components
 - Detail data flow and transformations
2. Error Handling:
 - Document all error conditions
 - Explain recovery procedures
 - Detail logging and reporting
3. API Interactions:
 - Document all platform method calls
 - Explain data transformations
 - Detail error handling
4. Financial Aid Impact:
 - Explain effects on aid calculations
 - Document regulatory compliance
 - Detail student eligibility impact
Example Section With All Aspects:
// CONDITIONALS:
// Check 1: Payment Period Status Validation
// Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.NfrPeriodsScript)
package CHANGE_ME // this should be updated to your actual package name
```
```

```
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
return acy
 .getTerms()
 .collect { period ->
 new NfrPeriod(period.number, period.name, period.startDate, period.endDate)
 }
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.

# NFR Awarding Criteria

Name: nfrAwardingCriteria

Purpose: Evaluates awarding criteria for an NFR period.

In Baseline Config: Yes

Return Type: Boolean

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "nfrAwardingCriteria"
Primary Function: Evaluates awarding criteria for an NFR period.
Expected Output Type: Boolean
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
periodNo
Purpose: Number of the NFR period being evaluated.
```

```
Type: Integer
Available Methods:
No methods available
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
```

```
    - Returns: LocalDate
    - Function: No description available'
  * getEnrollmentStatusSubType
    - Returns: String
    - Function: No description available'
  * getExternalProgramId
    - Returns: String
    - Function: No description available'
  * getFaEligibleIndicator
    - Returns: Boolean
    - Function: No description available'
  * getGpa
    - Returns: BigDecimal
    - Function: No description available'
  * getManualSapEvaluationIndicator
    - Returns: Boolean
    - Function: No description available'
  * getModality
    - Returns: String
    - Function: No description available'
  * getOfficialLastDateOfAttendance
    - Returns: LocalDate
    - Function: No description available'
  * getOpeId
    - Returns: String
    - Function: No description available'
  * getPrimaryLocation
    - Returns: String
    - Function: No description available'
  * getProgramLengthMeasurementUnit
    - Returns: String
    - Function: No description available'
  * getProgramLengthNumberOfUnits
    - Returns: int
    - Function: No description available'
  * getProgramMeasurementUnit
    - Returns: String
    - Function: No description available'
  * getProgramType
    - Returns: String
    - Function: No description available'
  * getRequiredGpa
    - Returns: BigDecimal
    - Function: No description available'
  * getSpecialPrograms
    - Returns: String
    - Function: No description available'
  * getStartDate
    - Returns: LocalDate
    - Function: No description available'
  * getStudentResources
    - Returns: List<IStudentResourceAPI>
    - Function: No description available'
  * getTermAcys
    - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
    - Function: No description available'
  * getTerms
    - Returns: ITermPeriodsAPI
    - Function: No description available'
  * getTermType
    - Returns: String
    - Function: No description available'
  * getTotalAcceptedTransferUnits
    - Returns: BigDecimal
    - Function: No description available'
  * getTotalRequiredUnits
```

```
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
```

```
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
```

```
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
Available Methods:
* getPeriodsIntersectingWithGroupForThePeriod
 - Returns: List<? extends IPeriod>
 - Function: Find the periods student was in the given group in the specified period of time'
* isInGroupAtAnyPointInThePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period at any point of time.'
* isInGroupAtDate
 - Returns: boolean
 - Function: Checks if student was in the given group on the given date.'
* isInGroupDuringCompletePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period for the whole time.'
* numberOfDaysInGroupForThePeriod
 - Returns: int
 - Function: Count number of days student was in the given group in the specified period of time'
* getActiveGroupsAtAnyPointInThePeriod
 - Returns: List<String>
 - Function: Find the student's groups active at any point during specified period of time'
* getActiveGroupsAtDate
 - Returns: List<String>
 - Function: Find the student's active groups at the given date'
* getActiveGroupsDuringCompletePeriod
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
acy
Purpose: The academic year of the funding element being evaluated.
Type: IAcademicYearScheduleAPI
Available Methods:
```

```
* getCompletedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getPeriodOfEnrollmentEndDate
 - Returns: LocalDate
 - Function: No description available'
* getPeriodOfEnrollmentStartDate
 - Returns: LocalDate
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
acyNo
Purpose: The number of the academic year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
acyOrder
Purpose: The order number of the academic year of the funding element being evaluated, among all non-
canceled academic
years.
Type: Integer
Available Methods:
No methods available
acys
Purpose: Provides access to all non-canceled academic years.
Type: IAcademicYearsScheduleAPI
Available Methods:
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
awy
Purpose: The award year of the funding element being evaluated.
Type: Integer
```

```
Available Methods:
No methods available
awyPeriod
Purpose: The period of the award year of the funding element being evaluated.
Type: IAwardYearAPI
Available Methods:
* asList
 - Returns: IAwardYearsAPI
 - Function: Returns an IAwardYearsAPI instance containing only this Award Year.'
* getEndYear
 - Returns: int
 - Function: Gets the end year of this Award Year, for example 2023 for Award Year 2022-2023.'
* getNext
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that follows this one chronologically.'
* getPrevious
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that precedes this one chronologically.'
* getStartYear
 - Returns: int
 - Function: Gets the start year of this Award Year, for example 2022 for Award Year 2022-2023.'
* getYear
 - Returns: int
 - Function: Short for getEndYear().'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
```

```
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
No methods available
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
```

```
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
```

```
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
```

```
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCode
Purpose: The fund code of the funding element being evaluated.
Type: String
Available Methods:
No methods available
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
fundType
```

```
Purpose: The fund type of the funding element being evaluated.
Type: String
Available Methods:
No methods available
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isFirstAcy
Purpose: True if the academic year of the funding element being evaluated is the first one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isLastAcy
Purpose: True if the academic year of the funding element being evaluated is the last one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
```

```
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
nfrConfig
Purpose: Interface that allows calling other NFR scripts for the same fund and student.
Type: INfrConfigAPI
Available Methods:
* evalAggregateLimit
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalAggregateLimit script for the same fund configuration and
 student as used
to evaluate the calling script.'
* evalAnnualLimit
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalAnnualLimit for the same fund configuration and student as used
 to
evaluate the calling script.'
* evalAwardingCriteria
 - Returns: Boolean
 - Function: Evaluate awardability of NFR fund for a given period number'
* evalDisbursementDate
 - Returns: LocalDate
 - Function: Equivalent to a call to the evalDisbursementDates script for the same fund configuration and
 student as
used to evaluate the calling script.'
* evalEligibility
 - Returns: INfrEligibilityResult
 - Function: Equivalent to a call to the evalEligibility script for the same fund configuration and student
 as used to
evaluate the calling script.'
* evalMaximumProjectedAwardAmount
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalMaximumProjectedAwardAmount script for the same fund
 configuration and
student as used to evaluate the calling script.'
* evalMinimumAmount
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalMinimumAmount script for the same fund configuration and
 student as used
to evaluate the calling script.'
* evalPeriods
 - Returns: IPeriods<NfrPeriod,?>
 - Function: Equivalent to a call to the evalPeriods script for the same fund configuration and student as
 used to evaluate the calling script.'
* getAwardYear
 - Returns: int
```

```
 - Function: Gets the award year of the configuration.'
* getAwardYearLimit
 - Returns: BigDecimal
 - Function: Gets the award year limit for the fund.'
nfrConfigs
Purpose: Maps (end years of) award years to the respective configuration in the current academic year
 calculation. Allows
calling other NFR scripts for the same fund and student but for different active award years in this
 academic year.
The user can rely on the entries in this map being ordered by award year.
Type: Map
Available Methods:
No methods available
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
```

```
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
 // Operation [number]: [name of calculation]
 // Formula: [complete mathematical expression]
 // Variables:
 // - [each variable and its purpose]
 // - [data types and units]
 // Steps:
 // 1. [detailed step-by-step breakdown]
 // 2. [intermediate calculations]
 // 3. [transformations applied]
 // Business Context: [why this calculation is needed]
 // Result: [meaning of the output]
 // Financial Impact: [effect on aid calculations]
 // Limitations: [any mathematical constraints]
 // If N/A, explicitly state: "ARITHMETIC: N/A"
ADDITIONAL REQUIREMENTS:
1. Complex Logic Sections:
 - Must break down multi-step operations
 - Explain interactions between components
 - Detail data flow and transformations
2. Error Handling:
 - Document all error conditions
 - Explain recovery procedures
 - Detail logging and reporting
3. API Interactions:
 - Document all platform method calls
 - Explain data transformations
 - Detail error handling
4. Financial Aid Impact:
 - Explain effects on aid calculations
 - Document regulatory compliance
 - Detail student eligibility impact
Example Section With All Aspects:
// CONDITIONALS:
// Check 1: Payment Period Status Validation
// Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
```

```
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.NfrAwardingCriteriaScript)
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
return true
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# NFR Disbursement Dates

Name: nfrDisbursementDates

Purpose: Evaluates the disbursement date for an NFR period.

In Baseline Config: Yes

Return Type: LocalDate

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "nfrDisbursementDates"
Primary Function: Evaluates the disbursement date for an NFR period.
Expected Output Type: LocalDate
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
```

```
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
```

```
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
Available Methods:
* getPeriodsIntersectingWithGroupForThePeriod
 - Returns: List<? extends IPeriod>
 - Function: Find the periods student was in the given group in the specified period of time'
* isInGroupAtAnyPointInThePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period at any point of time.'
* isInGroupAtDate
 - Returns: boolean
 - Function: Checks if student was in the given group on the given date.'
* isInGroupDuringCompletePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period for the whole time.'
* numberOfDaysInGroupForThePeriod
 - Returns: int
 - Function: Count number of days student was in the given group in the specified period of time'
* getActiveGroupsAtAnyPointInThePeriod
 - Returns: List<String>
 - Function: Find the student's groups active at any point during specified period of time'
* getActiveGroupsAtDate
 - Returns: List<String>
 - Function: Find the student's active groups at the given date'
* getActiveGroupsDuringCompletePeriod
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
acy
Purpose: The academic year of the funding element being evaluated.
Type: IAcademicYearScheduleAPI
Available Methods:
* getCompletedUnits
 - Returns: BigDecimal
```

```
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getPeriodOfEnrollmentEndDate
 - Returns: LocalDate
 - Function: No description available'
* getPeriodOfEnrollmentStartDate
 - Returns: LocalDate
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
acyNo
Purpose: The number of the academic year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
acyOrder
Purpose: The order number of the academic year of the funding element being evaluated, among all non-
canceled academic
years.
Type: Integer
Available Methods:
No methods available
acys
Purpose: Provides access to all non-canceled academic years.
Type: IAcademicYearsScheduleAPI
Available Methods:
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
awy
Purpose: The award year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
```

```
awyPeriod
Purpose: The period of the award year of the funding element being evaluated.
Type: IAwardYearAPI
Available Methods:
* asList
 - Returns: IAwardYearsAPI
 - Function: Returns an IAwardYearsAPI instance containing only this Award Year.'
* getEndYear
 - Returns: int
 - Function: Gets the end year of this Award Year, for example 2023 for Award Year 2022-2023.'
* getNext
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that follows this one chronologically.'
* getPrevious
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that precedes this one chronologically.'
* getStartYear
 - Returns: int
 - Function: Gets the start year of this Award Year, for example 2022 for Award Year 2022-2023.'
* getYear
 - Returns: int
 - Function: Short for getEndYear().'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
```

```
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
No methods available
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
```

```
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
```

```
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
```

```
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCode
Purpose: The fund code of the funding element being evaluated.
Type: String
Available Methods:
No methods available
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
fundType
Purpose: The fund type of the funding element being evaluated.
Type: String
```

```
Available Methods:
No methods available
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isFirstAcy
Purpose: True if the academic year of the funding element being evaluated is the first one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isLastAcy
Purpose: True if the academic year of the funding element being evaluated is the last one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
```

```
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
nfrConfig
Purpose: Interface that allows calling other NFR scripts for the same fund and student.
Type: INfrConfigAPI
Available Methods:
* evalAggregateLimit
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalAggregateLimit script for the same fund configuration and
 student as used
to evaluate the calling script.'
* evalAnnualLimit
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalAnnualLimit for the same fund configuration and student as used
 to
evaluate the calling script.'
* evalAwardingCriteria
 - Returns: Boolean
 - Function: Evaluate awardability of NFR fund for a given period number'
* evalDisbursementDate
 - Returns: LocalDate
 - Function: Equivalent to a call to the evalDisbursementDates script for the same fund configuration and
 student as
used to evaluate the calling script.'
* evalEligibility
 - Returns: INfrEligibilityResult
 - Function: Equivalent to a call to the evalEligibility script for the same fund configuration and student
 as used to
evaluate the calling script.'
* evalMaximumProjectedAwardAmount
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalMaximumProjectedAwardAmount script for the same fund
 configuration and
student as used to evaluate the calling script.'
* evalMinimumAmount
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalMinimumAmount script for the same fund configuration and
 student as used
to evaluate the calling script.'
* evalPeriods
 - Returns: IPeriods<NfrPeriod,?>
 - Function: Equivalent to a call to the evalPeriods script for the same fund configuration and student as
 used to evaluate the calling script.'
* getAwardYear
 - Returns: int
 - Function: Gets the award year of the configuration.'
* getAwardYearLimit
```

```
 - Returns: BigDecimal
 - Function: Gets the award year limit for the fund.'
nfrConfigs
Purpose: Maps (end years of) award years to the respective configuration in the current academic year
 calculation. Allows
calling other NFR scripts for the same fund and student but for different active award years in this
 academic year.
The user can rely on the entries in this map being ordered by award year.
Type: Map
Available Methods:
No methods available
period
Purpose: NFR period being evaluated.
Type: NfrPeriod
Available Methods:
No methods available
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
```

```
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
```

```
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
```

```
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
```

```
    // Source: [where boundary values come from]
    // Variables: [variables and methods involved]
    // Validation:
    // - [how boundaries are checked]
    // - [validation method details]
    // Edge Cases:
    // - [specific edge cases handled]
    // - [how each is managed]
    // Error Handling: [boundary violation management]
    // Impact: [effect of boundary violations]
    // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
  3. ARITHMETIC Section:
    Format:
    // ARITHMETIC:
    // Operation [number]: [name of calculation]
    // Formula: [complete mathematical expression]
    // Variables:
    // - [each variable and its purpose]
    // - [data types and units]
    // Steps:
    // 1. [detailed step-by-step breakdown]
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
  ADDITIONAL REQUIREMENTS:
  1. Complex Logic Sections:
    - Must break down multi-step operations
    - Explain interactions between components
    - Detail data flow and transformations
  2. Error Handling:
    - Document all error conditions
    - Explain recovery procedures
    - Detail logging and reporting
  3. API Interactions:
    - Document all platform method calls
    - Explain data transformations
    - Detail error handling
  4. Financial Aid Impact:
    - Explain effects on aid calculations
    - Document regulatory compliance
    - Detail student eligibility impact
  Example Section With All Aspects:
  // CONDITIONALS:
  // Check 1: Payment Period Status Validation
  // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
  // Purpose: Ensure only valid, active payment periods are included
  // Logic:
  // - Retrieve all loan payment periods
  // - Filter based on status (!= "CANCELED")
  // - Compare start dates with reference date
  // Success Path: Period included in calculations
  // Failure Path: Period excluded from totals
  // Business Rules: Canceled periods must not affect grade level
  // Error Handling: Invalid periods logged and skipped
  // Financial Impact: Affects available loan amounts
  //
  // BOUNDARY CHECKS:
  // Check 1: Payment Period Date Validation
  // Source: Academic Calendar Configuration
  // Variables: period.getStartDate(), referenceDate
  // Validation:
  // - Compare period start date with reference date
```

```
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.NfrDisbursementDatesScript)
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
import org.joda.time.LocalDate
//Schedule disbursement 5 days after the period started
return period.startDate.plusDays(5)
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# NFR Maximum Projected Award Amount

Name: nfrMaximumProjectedAwardAmount

Purpose: Evaluates the maximum projected award amount for an NFR period.

In Baseline Config: Yes

Return Type: BigDecimal

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
```

```
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "nfrMaximumProjectedAwardAmount"
Primary Function: Evaluates the maximum projected award amount for an NFR period.
Expected Output Type: BigDecimal
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
acy
Purpose: The academic year of the funding element being evaluated.
Type: IAcademicYearScheduleAPI
Available Methods:
* getCompletedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getPeriodOfEnrollmentEndDate
 - Returns: LocalDate
 - Function: No description available'
* getPeriodOfEnrollmentStartDate
 - Returns: LocalDate
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
acyNo
Purpose: The number of the academic year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
acyOrder
```

```
Purpose: The order number of the academic year of the funding element being evaluated, among all non-
canceled academic
years.
Type: Integer
Available Methods:
No methods available
acys
Purpose: Provides access to all non-canceled academic years.
Type: IAcademicYearsScheduleAPI
Available Methods:
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
awy
Purpose: The award year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
awyPeriod
Purpose: The period of the award year of the funding element being evaluated.
Type: IAwardYearAPI
Available Methods:
* asList
 - Returns: IAwardYearsAPI
 - Function: Returns an IAwardYearsAPI instance containing only this Award Year.'
* getEndYear
 - Returns: int
 - Function: Gets the end year of this Award Year, for example 2023 for Award Year 2022-2023.'
* getNext
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that follows this one chronologically.'
* getPrevious
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that precedes this one chronologically.'
* getStartYear
 - Returns: int
 - Function: Gets the start year of this Award Year, for example 2022 for Award Year 2022-2023.'
* getYear
 - Returns: int
 - Function: Short for getEndYear().'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
```

```
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
No methods available
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
```

```
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
```

```
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
```

```
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
```

```
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCode
Purpose: The fund code of the funding element being evaluated.
Type: String
Available Methods:
No methods available
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
fundType
Purpose: The fund type of the funding element being evaluated.
Type: String
Available Methods:
No methods available
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
```

```
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isFirstAcy
Purpose: True if the academic year of the funding element being evaluated is the first one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isLastAcy
Purpose: True if the academic year of the funding element being evaluated is the last one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
nfrConfig
Purpose: Interface that allows calling other NFR scripts for the same fund and student.
Type: INfrConfigAPI
Available Methods:
* evalAggregateLimit
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalAggregateLimit script for the same fund configuration and
 student as used
to evaluate the calling script.'
* evalAnnualLimit
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalAnnualLimit for the same fund configuration and student as used
 to
evaluate the calling script.'
* evalAwardingCriteria
 - Returns: Boolean
```

```
 - Function: Evaluate awardability of NFR fund for a given period number'
* evalDisbursementDate
 - Returns: LocalDate
 - Function: Equivalent to a call to the evalDisbursementDates script for the same fund configuration and
 student as
used to evaluate the calling script.'
* evalEligibility
 - Returns: INfrEligibilityResult
 - Function: Equivalent to a call to the evalEligibility script for the same fund configuration and student
 as used to
evaluate the calling script.'
* evalMaximumProjectedAwardAmount
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalMaximumProjectedAwardAmount script for the same fund
 configuration and
student as used to evaluate the calling script.'
* evalMinimumAmount
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalMinimumAmount script for the same fund configuration and
 student as used
to evaluate the calling script.'
* evalPeriods
 - Returns: IPeriods<NfrPeriod,?>
 - Function: Equivalent to a call to the evalPeriods script for the same fund configuration and student as
 used to evaluate the calling script.'
* getAwardYear
 - Returns: int
 - Function: Gets the award year of the configuration.'
* getAwardYearLimit
 - Returns: BigDecimal
 - Function: Gets the award year limit for the fund.'
nfrConfigs
Purpose: Maps (end years of) award years to the respective configuration in the current academic year
 calculation. Allows
calling other NFR scripts for the same fund and student but for different active award years in this
 academic year.
The user can rely on the entries in this map being ordered by award year.
Type: Map
Available Methods:
No methods available
period
Purpose: NFR period being evaluated.
Type: NfrPeriod
Available Methods:
No methods available
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
```

```
    - Returns: boolean
    - Function: No description available'
  * getAcademicCompletionDate
    - Returns: LocalDate
    - Function: No description available'
  * getAcademicStatus
    - Returns: String
    - Function: No description available'
  * getAdmissionStatus
    - Returns: String
    - Function: No description available'
  * getAssessedUnits
    - Returns: BigDecimal
    - Function: No description available'
  * getAydUnits
    - Returns: BigDecimal
    - Function: No description available'
  * getAydWeeks
    - Returns: Integer
    - Function: No description available'
  * getCharges
    - Returns: List<IChargeAPI>
    - Function: No description available'
  * getCode
    - Returns: String
    - Function: No description available'
  * getCollege
    - Returns: String
    - Function: No description available'
  * getCourses
    - Returns: ICoursesAPI
    - Function: No description available'
  * getDateOfDetermination
    - Returns: LocalDate
    - Function: No description available'
  * getDescription
    - Returns: String
    - Function: No description available'
  * getEnrollmentStatus
    - Returns: String
    - Function: No description available'
  * getEnrollmentStatusEffectiveDate
    - Returns: LocalDate
    - Function: No description available'
  * getEnrollmentStatusSubType
    - Returns: String
    - Function: No description available'
  * getExternalProgramId
    - Returns: String
    - Function: No description available'
  * getFaEligibleIndicator
    - Returns: Boolean
    - Function: No description available'
  * getGpa
    - Returns: BigDecimal
    - Function: No description available'
  * getManualSapEvaluationIndicator
    - Returns: Boolean
    - Function: No description available'
  * getModality
    - Returns: String
    - Function: No description available'
  * getOfficialLastDateOfAttendance
    - Returns: LocalDate
    - Function: No description available'
  * getOpeId
```

```
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
```

```
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
student
Purpose: Student information.
```

```
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
```

```
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
Available Methods:
* getPeriodsIntersectingWithGroupForThePeriod
 - Returns: List<? extends IPeriod>
 - Function: Find the periods student was in the given group in the specified period of time'
* isInGroupAtAnyPointInThePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period at any point of time.'
* isInGroupAtDate
 - Returns: boolean
 - Function: Checks if student was in the given group on the given date.'
* isInGroupDuringCompletePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period for the whole time.'
* numberOfDaysInGroupForThePeriod
 - Returns: int
 - Function: Count number of days student was in the given group in the specified period of time'
* getActiveGroupsAtAnyPointInThePeriod
 - Returns: List<String>
 - Function: Find the student's groups active at any point during specified period of time'
* getActiveGroupsAtDate
 - Returns: List<String>
 - Function: Find the student's active groups at the given date'
* getActiveGroupsDuringCompletePeriod
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
```

```
    // Purpose: [detailed business reason for this check]
    // Logic:
    // - [step-by-step breakdown of the condition]
    // - [how variables are used]
    // - [any transformations applied]
    // Success Path: [what happens when condition is true]
    // Failure Path: [what happens when condition is false]
    // Business Rules: [specific rules being enforced]
    // Error Handling: [how errors are managed]
    // Financial Impact: [effect on aid calculations]
    // If N/A, explicitly state: "CONDITIONALS: N/A"
   2. BOUNDARY CHECKS Section:
    Format:
    // BOUNDARY CHECKS:
    // Check [number]: [specific limit being enforced]
    // Source: [where boundary values come from]
    // Variables: [variables and methods involved]
    // Validation:
    // - [how boundaries are checked]
    // - [validation method details]
    // Edge Cases:
    // - [specific edge cases handled]
    // - [how each is managed]
    // Error Handling: [boundary violation management]
    // Impact: [effect of boundary violations]
    // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
   3. ARITHMETIC Section:
    Format:
    // ARITHMETIC:
    // Operation [number]: [name of calculation]
    // Formula: [complete mathematical expression]
    // Variables:
    // - [each variable and its purpose]
    // - [data types and units]
    // Steps:
    // 1. [detailed step-by-step breakdown]
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
   ADDITIONAL REQUIREMENTS:
   1. Complex Logic Sections:
    - Must break down multi-step operations
    - Explain interactions between components
    - Detail data flow and transformations
   2. Error Handling:
    - Document all error conditions
    - Explain recovery procedures
    - Detail logging and reporting
   3. API Interactions:
    - Document all platform method calls
    - Explain data transformations
    - Detail error handling
   4. Financial Aid Impact:
    - Explain effects on aid calculations
    - Document regulatory compliance
    - Detail student eligibility impact
   Example Section With All Aspects:
   // CONDITIONALS:
   // Check 1: Payment Period Status Validation
   // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
   // Purpose: Ensure only valid, active payment periods are included
   // Logic:
```

```
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.NfrMaximumProjectedAwardAmountScript)
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
return 2000
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# NFR Minimum Amount

Name: nfrMinimumAmount

Purpose: Evaluates the minimum amount for an NFR fund.

In Baseline Config: Yes

Return Type: BigDecimal

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "nfrMinimumAmount"
Primary Function: Evaluates the minimum amount for an NFR fund.
Expected Output Type: BigDecimal
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
acy
Purpose: The academic year of the funding element being evaluated.
Type: IAcademicYearScheduleAPI
Available Methods:
* getCompletedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getPeriodOfEnrollmentEndDate
 - Returns: LocalDate
```

```
 - Function: No description available'
* getPeriodOfEnrollmentStartDate
 - Returns: LocalDate
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
acyNo
Purpose: The number of the academic year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
acyOrder
Purpose: The order number of the academic year of the funding element being evaluated, among all non-
canceled academic
years.
Type: Integer
Available Methods:
No methods available
acys
Purpose: Provides access to all non-canceled academic years.
Type: IAcademicYearsScheduleAPI
Available Methods:
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
awy
Purpose: The award year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
awyPeriod
Purpose: The period of the award year of the funding element being evaluated.
Type: IAwardYearAPI
Available Methods:
* asList
 - Returns: IAwardYearsAPI
 - Function: Returns an IAwardYearsAPI instance containing only this Award Year.'
* getEndYear
 - Returns: int
 - Function: Gets the end year of this Award Year, for example 2023 for Award Year 2022-2023.'
* getNext
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that follows this one chronologically.'
* getPrevious
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that precedes this one chronologically.'
* getStartYear
 - Returns: int
 - Function: Gets the start year of this Award Year, for example 2022 for Award Year 2022-2023.'
* getYear
 - Returns: int
```

```
 - Function: Short for getEndYear().'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
No methods available
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
```

```
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
```

```
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
```

```
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
```

```
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCode
Purpose: The fund code of the funding element being evaluated.
Type: String
Available Methods:
No methods available
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
fundType
Purpose: The fund type of the funding element being evaluated.
Type: String
Available Methods:
No methods available
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
```

```
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isFirstAcy
Purpose: True if the academic year of the funding element being evaluated is the first one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isLastAcy
Purpose: True if the academic year of the funding element being evaluated is the last one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
```

```
nfrConfig
Purpose: Interface that allows calling other NFR scripts for the same fund and student.
Type: INfrConfigAPI
Available Methods:
* evalAggregateLimit
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalAggregateLimit script for the same fund configuration and
 student as used
to evaluate the calling script.'
* evalAnnualLimit
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalAnnualLimit for the same fund configuration and student as used
 to
evaluate the calling script.'
* evalAwardingCriteria
 - Returns: Boolean
 - Function: Evaluate awardability of NFR fund for a given period number'
* evalDisbursementDate
 - Returns: LocalDate
 - Function: Equivalent to a call to the evalDisbursementDates script for the same fund configuration and
 student as
used to evaluate the calling script.'
* evalEligibility
 - Returns: INfrEligibilityResult
 - Function: Equivalent to a call to the evalEligibility script for the same fund configuration and student
 as used to
evaluate the calling script.'
* evalMaximumProjectedAwardAmount
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalMaximumProjectedAwardAmount script for the same fund
 configuration and
student as used to evaluate the calling script.'
* evalMinimumAmount
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalMinimumAmount script for the same fund configuration and
 student as used
to evaluate the calling script.'
* evalPeriods
 - Returns: IPeriods<NfrPeriod,?>
 - Function: Equivalent to a call to the evalPeriods script for the same fund configuration and student as
 used to evaluate the calling script.'
* getAwardYear
 - Returns: int
 - Function: Gets the award year of the configuration.'
* getAwardYearLimit
 - Returns: BigDecimal
 - Function: Gets the award year limit for the fund.'
nfrConfigs
Purpose: Maps (end years of) award years to the respective configuration in the current academic year
 calculation. Allows
calling other NFR scripts for the same fund and student but for different active award years in this
 academic year.
The user can rely on the entries in this map being ordered by award year.
Type: Map
Available Methods:
No methods available
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
```

```
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
```

```
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
```

```
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
sap
Purpose: Information about the student's latest SAP.
```

```
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
```

```
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
Available Methods:
* getPeriodsIntersectingWithGroupForThePeriod
 - Returns: List<? extends IPeriod>
 - Function: Find the periods student was in the given group in the specified period of time'
* isInGroupAtAnyPointInThePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period at any point of time.'
* isInGroupAtDate
 - Returns: boolean
 - Function: Checks if student was in the given group on the given date.'
* isInGroupDuringCompletePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period for the whole time.'
* numberOfDaysInGroupForThePeriod
 - Returns: int
 - Function: Count number of days student was in the given group in the specified period of time'
* getActiveGroupsAtAnyPointInThePeriod
 - Returns: List<String>
 - Function: Find the student's groups active at any point during specified period of time'
* getActiveGroupsAtDate
 - Returns: List<String>
 - Function: Find the student's active groups at the given date'
* getActiveGroupsDuringCompletePeriod
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
```

```
   - Error handling approach
   - Integration points
   - Compliance considerations
  SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
  Every logical code section must include detailed comments for ALL three aspects:
  1. CONDITIONALS Section:
   Format:
   // CONDITIONALS:
   // Check [number]: [specific condition being evaluated]
   // Variables: [list of variables and methods involved]
   // Purpose: [detailed business reason for this check]
   // Logic:
   // - [step-by-step breakdown of the condition]
   // - [how variables are used]
   // - [any transformations applied]
   // Success Path: [what happens when condition is true]
   // Failure Path: [what happens when condition is false]
   // Business Rules: [specific rules being enforced]
   // Error Handling: [how errors are managed]
   // Financial Impact: [effect on aid calculations]
   // If N/A, explicitly state: "CONDITIONALS: N/A"
  2. BOUNDARY CHECKS Section:
   Format:
   // BOUNDARY CHECKS:
   // Check [number]: [specific limit being enforced]
   // Source: [where boundary values come from]
   // Variables: [variables and methods involved]
   // Validation:
   // - [how boundaries are checked]
   // - [validation method details]
   // Edge Cases:
   // - [specific edge cases handled]
   // - [how each is managed]
   // Error Handling: [boundary violation management]
   // Impact: [effect of boundary violations]
   // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
  3. ARITHMETIC Section:
   Format:
   // ARITHMETIC:
   // Operation [number]: [name of calculation]
   // Formula: [complete mathematical expression]
   // Variables:
   // - [each variable and its purpose]
   // - [data types and units]
   // Steps:
   // 1. [detailed step-by-step breakdown]
   // 2. [intermediate calculations]
   // 3. [transformations applied]
   // Business Context: [why this calculation is needed]
   // Result: [meaning of the output]
   // Financial Impact: [effect on aid calculations]
   // Limitations: [any mathematical constraints]
   // If N/A, explicitly state: "ARITHMETIC: N/A"
  ADDITIONAL REQUIREMENTS:
  1. Complex Logic Sections:
   - Must break down multi-step operations
   - Explain interactions between components
   - Detail data flow and transformations
  2. Error Handling:
   - Document all error conditions
   - Explain recovery procedures
   - Detail logging and reporting
  3. API Interactions:
   - Document all platform method calls
   - Explain data transformations
   - Detail error handling
```

```
4. Financial Aid Impact:
 - Explain effects on aid calculations
 - Document regulatory compliance
 - Detail student eligibility impact
Example Section With All Aspects:
// CONDITIONALS:
// Check 1: Payment Period Status Validation
// Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.NfrMinimumAmountScript)
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
1
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
```

```
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# NFR Annual Limit

Name: nfrAnnualLimit

Purpose: Evaluates the annual limit for an NFR fund.

In Baseline Config: Yes

Return Type: BigDecimal

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "nfrAnnualLimit"
Primary Function: Evaluates the annual limit for an NFR fund.
Expected Output Type: BigDecimal
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
acy
Purpose: The academic year of the funding element being evaluated.
Type: IAcademicYearScheduleAPI
Available Methods:
* getCompletedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
```

```
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getPeriodOfEnrollmentEndDate
 - Returns: LocalDate
 - Function: No description available'
* getPeriodOfEnrollmentStartDate
 - Returns: LocalDate
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
acyNo
Purpose: The number of the academic year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
acyOrder
Purpose: The order number of the academic year of the funding element being evaluated, among all non-
canceled academic
years.
Type: Integer
Available Methods:
No methods available
acys
Purpose: Provides access to all non-canceled academic years.
Type: IAcademicYearsScheduleAPI
Available Methods:
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
awy
Purpose: The award year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
awyPeriod
Purpose: The period of the award year of the funding element being evaluated.
Type: IAwardYearAPI
Available Methods:
* asList
 - Returns: IAwardYearsAPI
 - Function: Returns an IAwardYearsAPI instance containing only this Award Year.'
* getEndYear
 - Returns: int
 - Function: Gets the end year of this Award Year, for example 2023 for Award Year 2022-2023.'
* getNext
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that follows this one chronologically.'
* getPrevious
 - Returns: IAwardYearAPI
```

```
 - Function: Gets the Award Year that precedes this one chronologically.'
* getStartYear
 - Returns: int
 - Function: Gets the start year of this Award Year, for example 2022 for Award Year 2022-2023.'
* getYear
 - Returns: int
 - Function: Short for getEndYear().'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
```

```
Available Methods:
No methods available
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
```

```
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
```

```
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
```

```
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCode
Purpose: The fund code of the funding element being evaluated.
Type: String
Available Methods:
No methods available
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
fundType
Purpose: The fund type of the funding element being evaluated.
Type: String
Available Methods:
No methods available
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
```

```
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isFirstAcy
Purpose: True if the academic year of the funding element being evaluated is the first one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isLastAcy
Purpose: True if the academic year of the funding element being evaluated is the last one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
```

```
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
nfrConfig
Purpose: Interface that allows calling other NFR scripts for the same fund and student.
Type: INfrConfigAPI
Available Methods:
* evalAggregateLimit
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalAggregateLimit script for the same fund configuration and
 student as used
to evaluate the calling script.'
* evalAnnualLimit
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalAnnualLimit for the same fund configuration and student as used
 to
evaluate the calling script.'
* evalAwardingCriteria
 - Returns: Boolean
 - Function: Evaluate awardability of NFR fund for a given period number'
* evalDisbursementDate
 - Returns: LocalDate
 - Function: Equivalent to a call to the evalDisbursementDates script for the same fund configuration and
 student as
used to evaluate the calling script.'
* evalEligibility
 - Returns: INfrEligibilityResult
 - Function: Equivalent to a call to the evalEligibility script for the same fund configuration and student
 as used to
evaluate the calling script.'
* evalMaximumProjectedAwardAmount
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalMaximumProjectedAwardAmount script for the same fund
 configuration and
student as used to evaluate the calling script.'
* evalMinimumAmount
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalMinimumAmount script for the same fund configuration and
 student as used
to evaluate the calling script.'
* evalPeriods
 - Returns: IPeriods<NfrPeriod,?>
 - Function: Equivalent to a call to the evalPeriods script for the same fund configuration and student as
 used to evaluate the calling script.'
* getAwardYear
 - Returns: int
 - Function: Gets the award year of the configuration.'
* getAwardYearLimit
 - Returns: BigDecimal
 - Function: Gets the award year limit for the fund.'
nfrConfigs
Purpose: Maps (end years of) award years to the respective configuration in the current academic year
 calculation. Allows
calling other NFR scripts for the same fund and student but for different active award years in this
 academic year.
The user can rely on the entries in this map being ordered by award year.
Type: Map
Available Methods:
No methods available
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
```

```
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
```

```
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
```

```
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
```

```
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
```

```
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
Available Methods:
* getPeriodsIntersectingWithGroupForThePeriod
 - Returns: List<? extends IPeriod>
 - Function: Find the periods student was in the given group in the specified period of time'
* isInGroupAtAnyPointInThePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period at any point of time.'
* isInGroupAtDate
 - Returns: boolean
 - Function: Checks if student was in the given group on the given date.'
* isInGroupDuringCompletePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period for the whole time.'
* numberOfDaysInGroupForThePeriod
 - Returns: int
 - Function: Count number of days student was in the given group in the specified period of time'
* getActiveGroupsAtAnyPointInThePeriod
 - Returns: List<String>
 - Function: Find the student's groups active at any point during specified period of time'
* getActiveGroupsAtDate
 - Returns: List<String>
 - Function: Find the student's active groups at the given date'
* getActiveGroupsDuringCompletePeriod
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
```

```
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
 // Operation [number]: [name of calculation]
 // Formula: [complete mathematical expression]
 // Variables:
 // - [each variable and its purpose]
 // - [data types and units]
 // Steps:
 // 1. [detailed step-by-step breakdown]
 // 2. [intermediate calculations]
 // 3. [transformations applied]
 // Business Context: [why this calculation is needed]
 // Result: [meaning of the output]
 // Financial Impact: [effect on aid calculations]
 // Limitations: [any mathematical constraints]
 // If N/A, explicitly state: "ARITHMETIC: N/A"
ADDITIONAL REQUIREMENTS:
1. Complex Logic Sections:
 - Must break down multi-step operations
 - Explain interactions between components
 - Detail data flow and transformations
2. Error Handling:
 - Document all error conditions
```

```
   - Explain recovery procedures
   - Detail logging and reporting
  3. API Interactions:
   - Document all platform method calls
   - Explain data transformations
   - Detail error handling
  4. Financial Aid Impact:
   - Explain effects on aid calculations
   - Document regulatory compliance
   - Detail student eligibility impact
  Example Section With All Aspects:
  // CONDITIONALS:
  // Check 1: Payment Period Status Validation
  // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
  // Purpose: Ensure only valid, active payment periods are included
  // Logic:
  // - Retrieve all loan payment periods
  // - Filter based on status (!= "CANCELED")
  // - Compare start dates with reference date
  // Success Path: Period included in calculations
  // Failure Path: Period excluded from totals
  // Business Rules: Canceled periods must not affect grade level
  // Error Handling: Invalid periods logged and skipped
  // Financial Impact: Affects available loan amounts
  //
  // BOUNDARY CHECKS:
  // Check 1: Payment Period Date Validation
  // Source: Academic Calendar Configuration
  // Variables: period.getStartDate(), referenceDate
  // Validation:
  // - Compare period start date with reference date
  // - Ensure dates are not null
  // Edge Cases:
  // - Null dates are excluded
  // - Future dates are filtered
  // Error Handling: Invalid dates logged and skipped
  // Impact: Prevents future periods from affecting current calculation
  //
  // ARITHMETIC:
  // Operation 1: Payment Period Units Summation
  // Formula: sum(validPeriods.map(period => period.getUnits()))
  // Variables:
  // - period.getUnits(): Decimal, represents credit hours
  // - validPeriods: Array of filtered payment periods
  // Steps:
  // 1. Filter valid payment periods
  // 2. Extract units from each period
  // 3. Sum all units using reduce
  // Business Context: Calculate total completed units for grade level
  // Result: Total units affecting grade level
  // Financial Impact: Determines loan eligibility amounts
  // Limitations: Cannot exceed program maximum units
  ## Groovy Script for Analysis
  ```groovy
  //file:noinspection UnnecessaryQualifiedReference
  @groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.NfrAnnualLimitScript)
  package CHANGE_ME // this should be updated to your actual package name
  import com.oracle.sfp.scripting.api.*
  import com.oracle.sfp.scripting.api.util.*
  return 10000
  ```
  ## COMMENT GENERATION REQUIREMENTS:
  1. Add comprehensive comments following the exact format specified
  2. Every logical section must include all three aspects:
   - CONDITIONALS
   - BOUNDARY CHECKS
```

```
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# NFR Aggregate Limit

Name: nfrAggregateLimit

Purpose: Evaluates the aggregate limit for an NFR fund.

In Baseline Config: Yes

Return Type: BigDecimal

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "nfrAggregateLimit"
Primary Function: Evaluates the aggregate limit for an NFR fund.
Expected Output Type: BigDecimal
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
acy
Purpose: The academic year of the funding element being evaluated.
Type: IAcademicYearScheduleAPI
Available Methods:
* getCompletedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
```

```
 - Function: No description available'
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getPeriodOfEnrollmentEndDate
 - Returns: LocalDate
 - Function: No description available'
* getPeriodOfEnrollmentStartDate
 - Returns: LocalDate
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
acyNo
Purpose: The number of the academic year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
acyOrder
Purpose: The order number of the academic year of the funding element being evaluated, among all non-
canceled academic
years.
Type: Integer
Available Methods:
No methods available
acys
Purpose: Provides access to all non-canceled academic years.
Type: IAcademicYearsScheduleAPI
Available Methods:
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
awy
Purpose: The award year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
awyPeriod
Purpose: The period of the award year of the funding element being evaluated.
Type: IAwardYearAPI
Available Methods:
* asList
 - Returns: IAwardYearsAPI
 - Function: Returns an IAwardYearsAPI instance containing only this Award Year.'
* getEndYear
 - Returns: int
```

```
 - Function: Gets the end year of this Award Year, for example 2023 for Award Year 2022-2023.'
* getNext
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that follows this one chronologically.'
* getPrevious
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that precedes this one chronologically.'
* getStartYear
 - Returns: int
 - Function: Gets the start year of this Award Year, for example 2022 for Award Year 2022-2023.'
* getYear
 - Returns: int
 - Function: Short for getEndYear().'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
```

```
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
No methods available
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
```

```
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
```

```
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
```

```
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCode
Purpose: The fund code of the funding element being evaluated.
Type: String
Available Methods:
No methods available
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
fundType
Purpose: The fund type of the funding element being evaluated.
Type: String
Available Methods:
No methods available
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
```

```
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isFirstAcy
Purpose: True if the academic year of the funding element being evaluated is the first one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isLastAcy
Purpose: True if the academic year of the funding element being evaluated is the last one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
```

```
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
nfrConfig
Purpose: Interface that allows calling other NFR scripts for the same fund and student.
Type: INfrConfigAPI
Available Methods:
* evalAggregateLimit
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalAggregateLimit script for the same fund configuration and
 student as used
to evaluate the calling script.'
* evalAnnualLimit
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalAnnualLimit for the same fund configuration and student as used
 to
evaluate the calling script.'
* evalAwardingCriteria
 - Returns: Boolean
 - Function: Evaluate awardability of NFR fund for a given period number'
* evalDisbursementDate
 - Returns: LocalDate
 - Function: Equivalent to a call to the evalDisbursementDates script for the same fund configuration and
 student as
used to evaluate the calling script.'
* evalEligibility
 - Returns: INfrEligibilityResult
 - Function: Equivalent to a call to the evalEligibility script for the same fund configuration and student
 as used to
evaluate the calling script.'
* evalMaximumProjectedAwardAmount
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalMaximumProjectedAwardAmount script for the same fund
 configuration and
student as used to evaluate the calling script.'
* evalMinimumAmount
 - Returns: BigDecimal
 - Function: Equivalent to a call to the evalMinimumAmount script for the same fund configuration and
 student as used
to evaluate the calling script.'
* evalPeriods
 - Returns: IPeriods<NfrPeriod,?>
 - Function: Equivalent to a call to the evalPeriods script for the same fund configuration and student as
 used to evaluate the calling script.'
* getAwardYear
 - Returns: int
 - Function: Gets the award year of the configuration.'
* getAwardYearLimit
 - Returns: BigDecimal
 - Function: Gets the award year limit for the fund.'
nfrConfigs
Purpose: Maps (end years of) award years to the respective configuration in the current academic year
 calculation. Allows
calling other NFR scripts for the same fund and student but for different active award years in this
 academic year.
The user can rely on the entries in this map being ordered by award year.
Type: Map
```

```
Available Methods:
No methods available
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
```

```
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
```

```
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
```

```
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
```

```
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
Available Methods:
* getPeriodsIntersectingWithGroupForThePeriod
 - Returns: List<? extends IPeriod>
 - Function: Find the periods student was in the given group in the specified period of time'
* isInGroupAtAnyPointInThePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period at any point of time.'
* isInGroupAtDate
 - Returns: boolean
 - Function: Checks if student was in the given group on the given date.'
* isInGroupDuringCompletePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period for the whole time.'
* numberOfDaysInGroupForThePeriod
 - Returns: int
 - Function: Count number of days student was in the given group in the specified period of time'
* getActiveGroupsAtAnyPointInThePeriod
 - Returns: List<String>
 - Function: Find the student's groups active at any point during specified period of time'
* getActiveGroupsAtDate
 - Returns: List<String>
 - Function: Find the student's active groups at the given date'
* getActiveGroupsDuringCompletePeriod
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
```

```
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
 // Operation [number]: [name of calculation]
 // Formula: [complete mathematical expression]
 // Variables:
 // - [each variable and its purpose]
 // - [data types and units]
 // Steps:
 // 1. [detailed step-by-step breakdown]
 // 2. [intermediate calculations]
 // 3. [transformations applied]
 // Business Context: [why this calculation is needed]
 // Result: [meaning of the output]
 // Financial Impact: [effect on aid calculations]
 // Limitations: [any mathematical constraints]
 // If N/A, explicitly state: "ARITHMETIC: N/A"
ADDITIONAL REQUIREMENTS:
```

```
1. Complex Logic Sections:
 - Must break down multi-step operations
 - Explain interactions between components
 - Detail data flow and transformations
2. Error Handling:
 - Document all error conditions
 - Explain recovery procedures
 - Detail logging and reporting
3. API Interactions:
 - Document all platform method calls
 - Explain data transformations
 - Detail error handling
4. Financial Aid Impact:
 - Explain effects on aid calculations
 - Document regulatory compliance
 - Detail student eligibility impact
Example Section With All Aspects:
// CONDITIONALS:
// Check 1: Payment Period Status Validation
// Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.NfrAggregateLimitScript)
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
return 20000
```
```

```
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Satisfactory Academic Progress (SAP) Scripts

These are the prompts for scripts handling satisfactory academic progress calculations and evaluations.

## SAP Disable Pace of Completionv

Name: sapDisablePaceOfCompletion

Purpose: Determines if Pace of Completion calculations are needed for SAP.

In Baseline Config: No

Return Type: Boolean

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "sapDisablePaceOfCompletion"
Primary Function: Determines if the Pace Of Completion calculations needs to be considered for the SAP
 Period evaluation.
Expected Output Type: Boolean
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
```
```

```
When commenting the code, you MUST explain how these specific methods and bindings are being used:
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
No methods available
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
```

```
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
```

```
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
```

```
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
```

```
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
```

```
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
Available Methods:
* getPeriodsIntersectingWithGroupForThePeriod
 - Returns: List<? extends IPeriod>
 - Function: Find the periods student was in the given group in the specified period of time'
* isInGroupAtAnyPointInThePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period at any point of time.'
* isInGroupAtDate
 - Returns: boolean
 - Function: Checks if student was in the given group on the given date.'
* isInGroupDuringCompletePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period for the whole time.'
* numberOfDaysInGroupForThePeriod
 - Returns: int
 - Function: Count number of days student was in the given group in the specified period of time'
* getActiveGroupsAtAnyPointInThePeriod
 - Returns: List<String>
 - Function: Find the student's groups active at any point during specified period of time'
* getActiveGroupsAtDate
 - Returns: List<String>
```

```
 - Function: Find the student's active groups at the given date'
* getActiveGroupsDuringCompletePeriod
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
 // Operation [number]: [name of calculation]
 // Formula: [complete mathematical expression]
 // Variables:
 // - [each variable and its purpose]
 // - [data types and units]
 // Steps:
```

```
    // 1. [detailed step-by-step breakdown]
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
   ADDITIONAL REQUIREMENTS:
   1. Complex Logic Sections:
    - Must break down multi-step operations
    - Explain interactions between components
    - Detail data flow and transformations
   2. Error Handling:
    - Document all error conditions
    - Explain recovery procedures
    - Detail logging and reporting
   3. API Interactions:
    - Document all platform method calls
    - Explain data transformations
    - Detail error handling
   4. Financial Aid Impact:
    - Explain effects on aid calculations
    - Document regulatory compliance
    - Detail student eligibility impact
   Example Section With All Aspects:
   // CONDITIONALS:
   // Check 1: Payment Period Status Validation
   // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
   // Purpose: Ensure only valid, active payment periods are included
   // Logic:
   // - Retrieve all loan payment periods
   // - Filter based on status (!= "CANCELED")
   // - Compare start dates with reference date
   // Success Path: Period included in calculations
   // Failure Path: Period excluded from totals
   // Business Rules: Canceled periods must not affect grade level
   // Error Handling: Invalid periods logged and skipped
   // Financial Impact: Affects available loan amounts
   //
   // BOUNDARY CHECKS:
   // Check 1: Payment Period Date Validation
   // Source: Academic Calendar Configuration
   // Variables: period.getStartDate(), referenceDate
   // Validation:
   // - Compare period start date with reference date
   // - Ensure dates are not null
   // Edge Cases:
   // - Null dates are excluded
   // - Future dates are filtered
   // Error Handling: Invalid dates logged and skipped
   // Impact: Prevents future periods from affecting current calculation
   //
   // ARITHMETIC:
   // Operation 1: Payment Period Units Summation
   // Formula: sum(validPeriods.map(period => period.getUnits()))
   // Variables:
   // - period.getUnits(): Decimal, represents credit hours
   // - validPeriods: Array of filtered payment periods
   // Steps:
   // 1. Filter valid payment periods
   // 2. Extract units from each period
   // 3. Sum all units using reduce
   // Business Context: Calculate total completed units for grade level
   // Result: Total units affecting grade level
   // Financial Impact: Determines loan eligibility amounts
```

```
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
N/A - There is no baseline config. This is a placeholder.
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# SAP Evaluation Period Determination

Name: sapEvaluationPeriodDetermination

Purpose: Determines if SAP Period needs evaluation at end of academic year.

In Baseline Config: Yes

Return Type: Boolean

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "sapEvaluationPeriodDetermination"
Primary Function: Determines if the SAP Period needs to be evaluated at the end of the academic year.
Expected Output Type: Boolean
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
```

```
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
No methods available
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
```

```
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
```

```
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
```

```
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
```

```
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
```

```
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
Available Methods:
* getPeriodsIntersectingWithGroupForThePeriod
 - Returns: List<? extends IPeriod>
 - Function: Find the periods student was in the given group in the specified period of time'
* isInGroupAtAnyPointInThePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period at any point of time.'
* isInGroupAtDate
 - Returns: boolean
 - Function: Checks if student was in the given group on the given date.'
* isInGroupDuringCompletePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period for the whole time.'
* numberOfDaysInGroupForThePeriod
 - Returns: int
 - Function: Count number of days student was in the given group in the specified period of time'
* getActiveGroupsAtAnyPointInThePeriod
 - Returns: List<String>
 - Function: Find the student's groups active at any point during specified period of time'
* getActiveGroupsAtDate
 - Returns: List<String>
 - Function: Find the student's active groups at the given date'
* getActiveGroupsDuringCompletePeriod
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
```

```
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
 // Operation [number]: [name of calculation]
 // Formula: [complete mathematical expression]
 // Variables:
 // - [each variable and its purpose]
 // - [data types and units]
 // Steps:
 // 1. [detailed step-by-step breakdown]
 // 2. [intermediate calculations]
 // 3. [transformations applied]
 // Business Context: [why this calculation is needed]
 // Result: [meaning of the output]
 // Financial Impact: [effect on aid calculations]
```

```
 // Limitations: [any mathematical constraints]
 // If N/A, explicitly state: "ARITHMETIC: N/A"
ADDITIONAL REQUIREMENTS:
1. Complex Logic Sections:
 - Must break down multi-step operations
 - Explain interactions between components
 - Detail data flow and transformations
2. Error Handling:
 - Document all error conditions
 - Explain recovery procedures
 - Detail logging and reporting
3. API Interactions:
 - Document all platform method calls
 - Explain data transformations
 - Detail error handling
4. Financial Aid Impact:
 - Explain effects on aid calculations
 - Document regulatory compliance
 - Detail student eligibility impact
Example Section With All Aspects:
// CONDITIONALS:
// Check 1: Payment Period Status Validation
// Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.SapEvaluationPeriodDeterminationScript)
package CHANGE_ME // this should be updated to your actual package name
```
```

```
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
return false;
```
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Disbursement Scripts

These are the prompts for scripts controlling disbursement dates, criteria, and origination.

## Disbursement Dates

Name: disbursementDates

Purpose: Evaluates anticipated disbursement dates.

In Baseline Config: No

Return Type: List<LocalDate>

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "disbursementDates"
Primary Function: Evaluates anticipated disbursement dates for disbursement.
Expected Output Type: List<LocalDate>
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
```

```
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
acy
Purpose: The academic year of the funding element being evaluated.
Type: IAcademicYearScheduleAPI
Available Methods:
* getCompletedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getPeriodOfEnrollmentEndDate
 - Returns: LocalDate
 - Function: No description available'
* getPeriodOfEnrollmentStartDate
 - Returns: LocalDate
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
acyNo
Purpose: The number of the academic year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
acyOrder
Purpose: The order number of the academic year of the funding element being evaluated, among all non-
canceled academic
years.
Type: Integer
Available Methods:
No methods available
acys
Purpose: Provides access to all non-canceled academic years.
Type: IAcademicYearsScheduleAPI
Available Methods:
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
allAwardYearIsirs
```

```
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
awy
Purpose: The award year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
awyPeriod
Purpose: The period of the award year of the funding element being evaluated.
Type: IAwardYearAPI
Available Methods:
* asList
 - Returns: IAwardYearsAPI
 - Function: Returns an IAwardYearsAPI instance containing only this Award Year.'
* getEndYear
 - Returns: int
 - Function: Gets the end year of this Award Year, for example 2023 for Award Year 2022-2023.'
* getNext
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that follows this one chronologically.'
* getPrevious
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that precedes this one chronologically.'
* getStartYear
 - Returns: int
 - Function: Gets the start year of this Award Year, for example 2022 for Award Year 2022-2023.'
* getYear
 - Returns: int
 - Function: Short for getEndYear().'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
```

```
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
No methods available
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
```

```
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
```

```
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
```

```
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCode
Purpose: The fund code of the funding element being evaluated.
Type: String
Available Methods:
No methods available
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
```

```
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
fundType
Purpose: The fund type of the funding element being evaluated.
Type: String
Available Methods:
No methods available
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isFirstAcy
Purpose: True if the academic year of the funding element being evaluated is the first one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isFirstTimeBorrower
Purpose: Available for DLOANS. True if student is first time first year undergraduate borrower.
Type: Boolean
Available Methods:
No methods available
isLastAcy
Purpose: True if the academic year of the funding element being evaluated is the last one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
```

```
No methods available
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
lp
Purpose: The loan period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
lpNo
Purpose: The number of the loan period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
period
Purpose: A period matching the disbursement being evaluated.
Type: IPeriod
Available Methods:
No methods available
pp
Purpose: The payment period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
ppNo
Purpose: The number of the payment period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
previousFinancialPlan
Purpose: Provides access to previous financial plans.
```

```
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
```

```
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
```

```
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
```

```
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
scope
Purpose: Gathers all the scope details of the script being evaluated (acyNo, fundCode,
fundType, awy, lpNo, ppNo) allowing quick access to financialPlan/disbursements/etc
subsets directly pertaining to this scope.
Type: IFundScopeAPI
Available Methods:
* getAcyNo
 - Returns: int
 - Function: No description available'
* getAwy
 - Returns: int
 - Function: No description available'
* getFundCode
 - Returns: String
 - Function: No description available'
* getFundType
 - Returns: String
 - Function: No description available'
* getLpNo
 - Returns: int
 - Function: No description available'
* getPpNo
 - Returns: int
 - Function: No description available'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
```

```
    - Returns: Map<String,IReceivedDoc>
    - Function: Gets the student's documents, indexed by document code.'
  * getEmail
    - Returns: String
    - Function: Gets the e-mail.'
  * getEnrollmentStatus
    - Returns: String
    - Function: Gets the enrollment status.'
  * getExternalStudentId
    - Returns: String
    - Function: Gets the external student ID (unique per student).'
  * getFinancialAidRequested
    - Returns: Boolean
    - Function: Gets the financial aid request status.'
  * getFirstName
    - Returns: String
    - Function: Gets the first name.'
  * getGender
    - Returns: String
    - Function: Gets the gender.'
  * getHighSchoolEquivalency
    - Returns: Boolean
    - Function: Gets the high school equivalency status.'
  * getLastName
    - Returns: String
    - Function: Gets the last name.'
  * getMiddleName
    - Returns: String
    - Function: Gets the middle name.'
  * getPersonId
    - Returns: Long
    - Function: Gets the person ID.'
  * getProgramType
    - Returns: String
    - Function: Gets the program type.'
  * getRecentFinancialAidHistoryRecords
    - Returns: IRecentFahRecordsAPI
    - Function: Gets the recent financial aid records.'
  * getSchoolCode
    - Returns: String
    - Function: Gets the school code.'
  * getSchoolEmployee
    - Returns: Boolean
    - Function: Gets the school employee status.'
  * getSchoolEmployeeDependent
    - Returns: Boolean
    - Function: Gets the school employee dependent status.'
  * getSsn
    - Returns: ISocialSecurityNumber
    - Function: Gets the SSN.'
  * getStudentId
    - Returns: Long
    - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
Available Methods:
  * getPeriodsIntersectingWithGroupForThePeriod
    - Returns: List<? extends IPeriod>
    - Function: Find the periods student was in the given group in the specified period of time'
  * isInGroupAtAnyPointInThePeriod
    - Returns: boolean
    - Function: Checks if student was in the given group in the specified period at any point of time.'
  * isInGroupAtDate
    - Returns: boolean
    - Function: Checks if student was in the given group on the given date.'
```

```
* isInGroupDuringCompletePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period for the whole time.'
* numberOfDaysInGroupForThePeriod
 - Returns: int
 - Function: Count number of days student was in the given group in the specified period of time'
* getActiveGroupsAtAnyPointInThePeriod
 - Returns: List<String>
 - Function: Find the student's groups active at any point during specified period of time'
* getActiveGroupsAtDate
 - Returns: List<String>
 - Function: Find the student's active groups at the given date'
* getActiveGroupsDuringCompletePeriod
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
term
Purpose: The term of the funding element being evaluated.
Type: IAcademicTermPeriodAPI
Available Methods:
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getName
 - Returns: String
 - Function: No description available'
* isAttending
 - Returns: boolean
 - Function: No description available'
* isNotAttending
 - Returns: boolean
 - Function: No description available'
* isStandard
 - Returns: boolean
 - Function: No description available'
* isSummer
 - Returns: boolean
 - Function: No description available'
* isSummerHeader
 - Returns: boolean
 - Function: No description available'
* isSummerTrailer
 - Returns: boolean
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
termNo
Purpose: The number of the term of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
```

```
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
 // Operation [number]: [name of calculation]
 // Formula: [complete mathematical expression]
 // Variables:
 // - [each variable and its purpose]
 // - [data types and units]
 // Steps:
 // 1. [detailed step-by-step breakdown]
 // 2. [intermediate calculations]
 // 3. [transformations applied]
 // Business Context: [why this calculation is needed]
 // Result: [meaning of the output]
 // Financial Impact: [effect on aid calculations]
 // Limitations: [any mathematical constraints]
 // If N/A, explicitly state: "ARITHMETIC: N/A"
ADDITIONAL REQUIREMENTS:
1. Complex Logic Sections:
 - Must break down multi-step operations
 - Explain interactions between components
 - Detail data flow and transformations
2. Error Handling:
 - Document all error conditions
 - Explain recovery procedures
```

```
 - Detail logging and reporting
3. API Interactions:
 - Document all platform method calls
 - Explain data transformations
 - Detail error handling
4. Financial Aid Impact:
 - Explain effects on aid calculations
 - Document regulatory compliance
 - Detail student eligibility impact
Example Section With All Aspects:
// CONDITIONALS:
// Check 1: Payment Period Status Validation
// Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
N/A - There is no baseline config. This is a placeholder.
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
```

```
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Configurable Disbursement Criteria

Name: configurableDisbursementCriteria

Purpose: Evaluates configurable disbursement criteria.

In Baseline Config: Yes

Return Type: Boolean

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "configurableDisbursementCriteria"
Primary Function: Evaluates configurable disbursement criteria.
Expected Output Type: Boolean
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
acy
Purpose: The academic year of the funding element being evaluated.
Type: IAcademicYearScheduleAPI
Available Methods:
* getCompletedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
```

```
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getPeriodOfEnrollmentEndDate
 - Returns: LocalDate
 - Function: No description available'
* getPeriodOfEnrollmentStartDate
 - Returns: LocalDate
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
acyNo
Purpose: The number of the academic year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
acyOrder
Purpose: The order number of the academic year of the funding element being evaluated, among all non-
canceled academic
years.
Type: Integer
Available Methods:
No methods available
acys
Purpose: Provides access to all non-canceled academic years.
Type: IAcademicYearsScheduleAPI
Available Methods:
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
awy
Purpose: The award year of the disbursement being evaluated.
Type: Integer
Available Methods:
No methods available
awyPeriod
Purpose: The period of the award year of the funding element being evaluated.
Type: IAwardYearAPI
Available Methods:
* asList
 - Returns: IAwardYearsAPI
 - Function: Returns an IAwardYearsAPI instance containing only this Award Year.'
* getEndYear
 - Returns: int
 - Function: Gets the end year of this Award Year, for example 2023 for Award Year 2022-2023.'
* getNext
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that follows this one chronologically.'
* getPrevious
 - Returns: IAwardYearAPI
```

```
 - Function: Gets the Award Year that precedes this one chronologically.'
* getStartYear
 - Returns: int
 - Function: Gets the start year of this Award Year, for example 2022 for Award Year 2022-2023.'
* getYear
 - Returns: int
 - Function: Short for getEndYear().'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
```

```
Available Methods:
No methods available
currentDisbursementDate
Purpose: The anticipated disbursement date of the disbursement being evaluated.
Type: LocalDate
Available Methods:
No methods available
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
```

```
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
```

```
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
```

```
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCode
Purpose: The fund code of the funding element being evaluated.
Type: String
Available Methods:
No methods available
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
fundType
Purpose: The fund type of the funding element being evaluated.
Type: String
Available Methods:
No methods available
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
```

```
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isFirstAcy
Purpose: True if the academic year of the funding element being evaluated is the first one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isLastAcy
Purpose: True if the academic year of the funding element being evaluated is the last one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
```

```
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
lp
Purpose: The loan period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
lpNo
Purpose: The number of the loan period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
period
Purpose: The period associated with the disbursement being evaluated.
Type: IDisbursementPeriodAPI
Available Methods:
* getAcademicYearNumber
 - Returns: int
 - Function: No description available'
* getNumberOfCredits
 - Returns: int
 - Function: No description available'
* getPeriodType
 - Returns: String
 - Function: No description available'
* isGrantPaymentPeriod
 - Returns: boolean
 - Function: No description available'
* isLoanPaymentPeriod
 - Returns: boolean
 - Function: No description available'
* isNfrPeriod
 - Returns: boolean
 - Function: No description available'
pp
Purpose: The payment period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
ppNo
Purpose: The number of the payment period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
```

```
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
```

```
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
```

```
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
```

```
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
scope
Purpose: Gathers all the scope details of the script being evaluated (acyNo, fundCode,
fundType, awy, lpNo, ppNo) allowing quick access to financialPlan/disbursements/etc
subsets directly pertaining to this scope.
Type: IFundScopeAPI
Available Methods:
* getAcyNo
 - Returns: int
 - Function: No description available'
* getAwy
 - Returns: int
 - Function: No description available'
* getFundCode
 - Returns: String
 - Function: No description available'
* getFundType
 - Returns: String
 - Function: No description available'
* getLpNo
 - Returns: int
 - Function: No description available'
* getPpNo
 - Returns: int
 - Function: No description available'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
```

```
    - Returns: String
    - Function: Gets the enrollment status.'
  * getExternalStudentId
    - Returns: String
    - Function: Gets the external student ID (unique per student).'
  * getFinancialAidRequested
    - Returns: Boolean
    - Function: Gets the financial aid request status.'
  * getFirstName
    - Returns: String
    - Function: Gets the first name.'
  * getGender
    - Returns: String
    - Function: Gets the gender.'
  * getHighSchoolEquivalency
    - Returns: Boolean
    - Function: Gets the high school equivalency status.'
  * getLastName
    - Returns: String
    - Function: Gets the last name.'
  * getMiddleName
    - Returns: String
    - Function: Gets the middle name.'
  * getPersonId
    - Returns: Long
    - Function: Gets the person ID.'
  * getProgramType
    - Returns: String
    - Function: Gets the program type.'
  * getRecentFinancialAidHistoryRecords
    - Returns: IRecentFahRecordsAPI
    - Function: Gets the recent financial aid records.'
  * getSchoolCode
    - Returns: String
    - Function: Gets the school code.'
  * getSchoolEmployee
    - Returns: Boolean
    - Function: Gets the school employee status.'
  * getSchoolEmployeeDependent
    - Returns: Boolean
    - Function: Gets the school employee dependent status.'
  * getSsn
    - Returns: ISocialSecurityNumber
    - Function: Gets the SSN.'
  * getStudentId
    - Returns: Long
    - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
Available Methods:
  * getPeriodsIntersectingWithGroupForThePeriod
    - Returns: List<? extends IPeriod>
    - Function: Find the periods student was in the given group in the specified period of time'
  * isInGroupAtAnyPointInThePeriod
    - Returns: boolean
    - Function: Checks if student was in the given group in the specified period at any point of time.'
  * isInGroupAtDate
    - Returns: boolean
    - Function: Checks if student was in the given group on the given date.'
  * isInGroupDuringCompletePeriod
    - Returns: boolean
    - Function: Checks if student was in the given group in the specified period for the whole time.'
  * numberOfDaysInGroupForThePeriod
    - Returns: int
    - Function: Count number of days student was in the given group in the specified period of time'
```

```
* getActiveGroupsAtAnyPointInThePeriod
 - Returns: List<String>
 - Function: Find the student's groups active at any point during specified period of time'
* getActiveGroupsAtDate
 - Returns: List<String>
 - Function: Find the student's active groups at the given date'
* getActiveGroupsDuringCompletePeriod
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
term
Purpose: The term of the funding element being evaluated.
Type: IAcademicTermPeriodAPI
Available Methods:
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getName
 - Returns: String
 - Function: No description available'
* isAttending
 - Returns: boolean
 - Function: No description available'
* isNotAttending
 - Returns: boolean
 - Function: No description available'
* isStandard
 - Returns: boolean
 - Function: No description available'
* isSummer
 - Returns: boolean
 - Function: No description available'
* isSummerHeader
 - Returns: boolean
 - Function: No description available'
* isSummerTrailer
 - Returns: boolean
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
termNo
Purpose: The number of the term of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
```

```
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
 // Operation [number]: [name of calculation]
 // Formula: [complete mathematical expression]
 // Variables:
 // - [each variable and its purpose]
 // - [data types and units]
 // Steps:
 // 1. [detailed step-by-step breakdown]
 // 2. [intermediate calculations]
 // 3. [transformations applied]
 // Business Context: [why this calculation is needed]
 // Result: [meaning of the output]
 // Financial Impact: [effect on aid calculations]
 // Limitations: [any mathematical constraints]
 // If N/A, explicitly state: "ARITHMETIC: N/A"
ADDITIONAL REQUIREMENTS:
1. Complex Logic Sections:
 - Must break down multi-step operations
 - Explain interactions between components
 - Detail data flow and transformations
2. Error Handling:
 - Document all error conditions
 - Explain recovery procedures
 - Detail logging and reporting
3. API Interactions:
 - Document all platform method calls
 - Explain data transformations
 - Detail error handling
4. Financial Aid Impact:
```

```
 - Explain effects on aid calculations
 - Document regulatory compliance
 - Detail student eligibility impact
Example Section With All Aspects:
// CONDITIONALS:
// Check 1: Payment Period Status Validation
// Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.ConfigurableDisbursementCriteriaScript)
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
return cod.getCrecs().firstMatch({ crec -> crec.getCounselingCompleteDate() != null; }).isPresent();
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
```

```
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Configurable Late Disbursement Criteria

Name: configurableLateDisbursementCriteria

Purpose: Evaluates configurable late/pwd disbursement criteria.

In Baseline Config: No

Return Type: Boolean

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "configurableLateDisbursementCriteria"
Primary Function: Evaluates configurable late/pwd disbursement criteria.
Expected Output Type: Boolean
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
acy
Purpose: The academic year of the funding element being evaluated.
Type: IAcademicYearScheduleAPI
Available Methods:
* getCompletedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
```

```
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getPeriodOfEnrollmentEndDate
 - Returns: LocalDate
 - Function: No description available'
* getPeriodOfEnrollmentStartDate
 - Returns: LocalDate
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
acyNo
Purpose: The number of the academic year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
acyOrder
Purpose: The order number of the academic year of the funding element being evaluated, among all non-
canceled academic
years.
Type: Integer
Available Methods:
No methods available
acys
Purpose: Provides access to all non-canceled academic years.
Type: IAcademicYearsScheduleAPI
Available Methods:
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
awy
Purpose: The award year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
awyPeriod
Purpose: The period of the award year of the funding element being evaluated.
Type: IAwardYearAPI
Available Methods:
* asList
 - Returns: IAwardYearsAPI
 - Function: Returns an IAwardYearsAPI instance containing only this Award Year.'
* getEndYear
 - Returns: int
 - Function: Gets the end year of this Award Year, for example 2023 for Award Year 2022-2023.'
* getNext
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that follows this one chronologically.'
* getPrevious
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that precedes this one chronologically.'
```

```
* getStartYear
 - Returns: int
 - Function: Gets the start year of this Award Year, for example 2022 for Award Year 2022-2023.'
* getYear
 - Returns: int
 - Function: Short for getEndYear().'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
```

```
No methods available
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
```

```
    - Returns: T
    - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
    status.'
  * getWithStatuses
    - Returns: T
    - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
    statuses.'
  * hasBeenCancelled
    - Returns: boolean
    - Function: No description available'
  * isDisbursed
    - Returns: boolean
    - Function: No description available'
  * isLateDisbursement
    - Returns: boolean
    - Function: No description available'
  * isLockingPp
    - Returns: boolean
    - Function: No description available'
  * isPwd
    - Returns: boolean
    - Function: No description available'
  * getPpMaxDisbursementAmount
    - Returns: BigDecimal
    - Function: No description available'
  * getPpTotalDecreasedAmount
    - Returns: BigDecimal
    - Function: No description available'
  dloan
  Purpose: Information related to Direct Loans.
  Type: IDloanAPI
  Available Methods:
  * getLifetimeLimit
    - Returns: int
    - Function: No description available'
  * getRemainingNsldsLimit
    - Returns: int
    - Function: No description available'
  enrollment
  Purpose: Information for the enrollment this script is being evaluated for.
  Type: IEnrollmentAPI
  Available Methods:
  * belongsTo
    - Returns: boolean
    - Function: Checks if this enrollment belongs to a student.'
  * getBindings
    - Returns: IEnrollmentBindingsAPI
    - Function: Gets bindings for this enrollment.'
  * getEnrollmentStatus
    - Returns: String
    - Function: Gets the enrollment status.'
  * getProgramStartDate
    - Returns: LocalDate
    - Function: Gets the program start date.'
  * isEffective
    - Returns: boolean
    - Function: Checks if this enrollment is effective.'
  enrollments
  Purpose: Enrollments information.
  Type: IEnrollmentsAPI
  Available Methods:
  * getHigherPriorityThan
    - Returns: List<IEnrollmentAPI>
    - Function: Gets a list of enrollments with higher priority than a given enrollment.'
  * getOtherThan
    - Returns: List<IEnrollmentAPI>
```

```
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
```

```
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCode
Purpose: The fund code of the funding element being evaluated.
Type: String
Available Methods:
No methods available
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
fundType
Purpose: The fund type of the funding element being evaluated.
Type: String
Available Methods:
No methods available
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
```

```
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isFirstAcy
Purpose: True if the academic year of the funding element being evaluated is the first one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isLastAcy
Purpose: True if the academic year of the funding element being evaluated is the last one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
```

```
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
lp
Purpose: The loan period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
lpNo
Purpose: The number of the loan period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
period
Purpose: The period associated with the late/pwd disbursement being evaluated.
Type: IDisbursementPeriodAPI
Available Methods:
* getAcademicYearNumber
 - Returns: int
 - Function: No description available'
* getNumberOfCredits
 - Returns: int
 - Function: No description available'
* getPeriodType
 - Returns: String
 - Function: No description available'
* isGrantPaymentPeriod
 - Returns: boolean
 - Function: No description available'
* isLoanPaymentPeriod
 - Returns: boolean
 - Function: No description available'
* isNfrPeriod
 - Returns: boolean
 - Function: No description available'
pp
Purpose: The payment period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
ppNo
Purpose: The number of the payment period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
```

```
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
```

```
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
```

```
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
```

```
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
scope
Purpose: Gathers all the scope details of the script being evaluated (acyNo, fundCode,
fundType, awy, lpNo, ppNo) allowing quick access to financialPlan/disbursements/etc
subsets directly pertaining to this scope.
Type: IFundScopeAPI
Available Methods:
* getAcyNo
 - Returns: int
 - Function: No description available'
* getAwy
 - Returns: int
 - Function: No description available'
* getFundCode
 - Returns: String
 - Function: No description available'
* getFundType
 - Returns: String
 - Function: No description available'
* getLpNo
 - Returns: int
 - Function: No description available'
* getPpNo
 - Returns: int
 - Function: No description available'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
```

```
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
Available Methods:
* getPeriodsIntersectingWithGroupForThePeriod
 - Returns: List<? extends IPeriod>
 - Function: Find the periods student was in the given group in the specified period of time'
* isInGroupAtAnyPointInThePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period at any point of time.'
* isInGroupAtDate
 - Returns: boolean
 - Function: Checks if student was in the given group on the given date.'
* isInGroupDuringCompletePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period for the whole time.'
* numberOfDaysInGroupForThePeriod
 - Returns: int
 - Function: Count number of days student was in the given group in the specified period of time'
* getActiveGroupsAtAnyPointInThePeriod
 - Returns: List<String>
 - Function: Find the student's groups active at any point during specified period of time'
* getActiveGroupsAtDate
 - Returns: List<String>
 - Function: Find the student's active groups at the given date'
* getActiveGroupsDuringCompletePeriod
```

```
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
term
Purpose: The term of the funding element being evaluated.
Type: IAcademicTermPeriodAPI
Available Methods:
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getName
 - Returns: String
 - Function: No description available'
* isAttending
 - Returns: boolean
 - Function: No description available'
* isNotAttending
 - Returns: boolean
 - Function: No description available'
* isStandard
 - Returns: boolean
 - Function: No description available'
* isSummer
 - Returns: boolean
 - Function: No description available'
* isSummerHeader
 - Returns: boolean
 - Function: No description available'
* isSummerTrailer
 - Returns: boolean
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
termNo
Purpose: The number of the term of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
```

```
    // Check [number]: [specific condition being evaluated]
    // Variables: [list of variables and methods involved]
    // Purpose: [detailed business reason for this check]
    // Logic:
    // - [step-by-step breakdown of the condition]
    // - [how variables are used]
    // - [any transformations applied]
    // Success Path: [what happens when condition is true]
    // Failure Path: [what happens when condition is false]
    // Business Rules: [specific rules being enforced]
    // Error Handling: [how errors are managed]
    // Financial Impact: [effect on aid calculations]
    // If N/A, explicitly state: "CONDITIONALS: N/A"
  2. BOUNDARY CHECKS Section:
    Format:
    // BOUNDARY CHECKS:
    // Check [number]: [specific limit being enforced]
    // Source: [where boundary values come from]
    // Variables: [variables and methods involved]
    // Validation:
    // - [how boundaries are checked]
    // - [validation method details]
    // Edge Cases:
    // - [specific edge cases handled]
    // - [how each is managed]
    // Error Handling: [boundary violation management]
    // Impact: [effect of boundary violations]
    // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
  3. ARITHMETIC Section:
    Format:
    // ARITHMETIC:
    // Operation [number]: [name of calculation]
    // Formula: [complete mathematical expression]
    // Variables:
    // - [each variable and its purpose]
    // - [data types and units]
    // Steps:
    // 1. [detailed step-by-step breakdown]
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
  ADDITIONAL REQUIREMENTS:
  1. Complex Logic Sections:
    - Must break down multi-step operations
    - Explain interactions between components
    - Detail data flow and transformations
  2. Error Handling:
    - Document all error conditions
    - Explain recovery procedures
    - Detail logging and reporting
  3. API Interactions:
    - Document all platform method calls
    - Explain data transformations
    - Detail error handling
  4. Financial Aid Impact:
    - Explain effects on aid calculations
    - Document regulatory compliance
    - Detail student eligibility impact
  Example Section With All Aspects:
  // CONDITIONALS:
  // Check 1: Payment Period Status Validation
  // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
```

```
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
N/A - There is no baseline config. This is a placeholder.
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Origination Criteria

Name: originationCriteria

Purpose: Evaluates origination criteria for a disbursement.

In Baseline Config: Yes

Return Type: Boolean

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "originationCriteria"
Primary Function: Evaluates origination criteria for a disbursement.
Expected Output Type: Boolean
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
acy
Purpose: The academic year of the funding element being evaluated.
Type: IAcademicYearScheduleAPI
Available Methods:
* getCompletedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getPeriodOfEnrollmentEndDate
 - Returns: LocalDate
 - Function: No description available'
* getPeriodOfEnrollmentStartDate
 - Returns: LocalDate
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
```

```
 - Function: No description available'
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
acyNo
Purpose: The number of the academic year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
acyOrder
Purpose: The order number of the academic year of the funding element being evaluated, among all non-
canceled academic
years.
Type: Integer
Available Methods:
No methods available
acys
Purpose: Provides access to all non-canceled academic years.
Type: IAcademicYearsScheduleAPI
Available Methods:
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
awy
Purpose: The award year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
awyPeriod
Purpose: The period of the award year of the funding element being evaluated.
Type: IAwardYearAPI
Available Methods:
* asList
 - Returns: IAwardYearsAPI
 - Function: Returns an IAwardYearsAPI instance containing only this Award Year.'
* getEndYear
 - Returns: int
 - Function: Gets the end year of this Award Year, for example 2023 for Award Year 2022-2023.'
* getNext
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that follows this one chronologically.'
* getPrevious
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that precedes this one chronologically.'
* getStartYear
 - Returns: int
 - Function: Gets the start year of this Award Year, for example 2022 for Award Year 2022-2023.'
* getYear
 - Returns: int
 - Function: Short for getEndYear().'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
```

```
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
No methods available
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
```

```
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
```

```
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
financialPlan
Purpose: Information about the student's current financial plan.
```

```
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
```

```
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCode
Purpose: The fund code of the funding element being evaluated.
Type: String
Available Methods:
No methods available
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
fundType
Purpose: The fund type of the funding element being evaluated.
Type: String
Available Methods:
No methods available
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
```

```
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isFirstAcy
Purpose: True if the academic year of the funding element being evaluated is the first one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isLastAcy
Purpose: True if the academic year of the funding element being evaluated is the last one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
lp
Purpose: The loan period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
```

```
 - Function: No description available'
lpNo
Purpose: The number of the loan period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
pp
Purpose: The payment period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
ppNo
Purpose: The number of the payment period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
```

```
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
```

```
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
```

```
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
scope
Purpose: Gathers all the scope details of the script being evaluated (acyNo, fundCode,
fundType, awy, lpNo, ppNo) allowing quick access to financialPlan/disbursements/etc
subsets directly pertaining to this scope.
Type: IFundScopeAPI
Available Methods:
* getAcyNo
 - Returns: int
 - Function: No description available'
* getAwy
 - Returns: int
 - Function: No description available'
* getFundCode
 - Returns: String
 - Function: No description available'
* getFundType
 - Returns: String
 - Function: No description available'
* getLpNo
 - Returns: int
 - Function: No description available'
* getPpNo
 - Returns: int
 - Function: No description available'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
```

```
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
```

```
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
Available Methods:
* getPeriodsIntersectingWithGroupForThePeriod
 - Returns: List<? extends IPeriod>
 - Function: Find the periods student was in the given group in the specified period of time'
* isInGroupAtAnyPointInThePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period at any point of time.'
* isInGroupAtDate
 - Returns: boolean
 - Function: Checks if student was in the given group on the given date.'
* isInGroupDuringCompletePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period for the whole time.'
* numberOfDaysInGroupForThePeriod
 - Returns: int
 - Function: Count number of days student was in the given group in the specified period of time'
* getActiveGroupsAtAnyPointInThePeriod
 - Returns: List<String>
 - Function: Find the student's groups active at any point during specified period of time'
* getActiveGroupsAtDate
 - Returns: List<String>
 - Function: Find the student's active groups at the given date'
* getActiveGroupsDuringCompletePeriod
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
term
Purpose: The term of the funding element being evaluated.
Type: IAcademicTermPeriodAPI
Available Methods:
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getName
 - Returns: String
 - Function: No description available'
* isAttending
 - Returns: boolean
 - Function: No description available'
* isNotAttending
 - Returns: boolean
 - Function: No description available'
* isStandard
 - Returns: boolean
 - Function: No description available'
* isSummer
 - Returns: boolean
 - Function: No description available'
* isSummerHeader
 - Returns: boolean
 - Function: No description available'
* isSummerTrailer
 - Returns: boolean
 - Function: No description available'
```

```
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
termNo
Purpose: The number of the term of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
 // Operation [number]: [name of calculation]
 // Formula: [complete mathematical expression]
```

```
    // Variables:
    // - [each variable and its purpose]
    // - [data types and units]
    // Steps:
    // 1. [detailed step-by-step breakdown]
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
ADDITIONAL REQUIREMENTS:
1. Complex Logic Sections:
 - Must break down multi-step operations
 - Explain interactions between components
 - Detail data flow and transformations
2. Error Handling:
 - Document all error conditions
 - Explain recovery procedures
 - Detail logging and reporting
3. API Interactions:
 - Document all platform method calls
 - Explain data transformations
 - Detail error handling
4. Financial Aid Impact:
 - Explain effects on aid calculations
 - Document regulatory compliance
 - Detail student eligibility impact
Example Section With All Aspects:
// CONDITIONALS:
// Check 1: Payment Period Status Validation
// Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
```

```
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.OriginationCriteriaScript)
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
return true;
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Miscellaneous Scripts

These are the prompts for utility scripts that handle various platform functions.

## Letter Code Criteria

Name: letterCodeCriteria

Purpose: Determines the letter code for given parameters.

In Baseline Config: Yes

Return Type: String

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "letterCodeCriteria"
Primary Function: Determines the letter code for a given set of parameters.
Expected Output Type: String
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
```

```
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
letterCode
Purpose: Binding that encapsulates parameters used to determine letters codes.
Type: ILetterCodeAPI
Available Methods:
* findLastPublished
 - Returns: ILetterAPI
 - Function: Finds the most recent letter published for the current set
of identifying parameters and respective letter code.'
* getParams
 - Returns: @NonNull Map<String,String>
 - Function: Gets the letter parameters.'
* querySql
 - Returns: String
 - Function: Executes a SQL Query against the FASRPT schema and returns a string.'
log
Purpose: Provides logging capability to scripts.
```

```
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
```

```
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
 // Operation [number]: [name of calculation]
 // Formula: [complete mathematical expression]
 // Variables:
 // - [each variable and its purpose]
 // - [data types and units]
 // Steps:
 // 1. [detailed step-by-step breakdown]
 // 2. [intermediate calculations]
 // 3. [transformations applied]
 // Business Context: [why this calculation is needed]
 // Result: [meaning of the output]
 // Financial Impact: [effect on aid calculations]
 // Limitations: [any mathematical constraints]
 // If N/A, explicitly state: "ARITHMETIC: N/A"
ADDITIONAL REQUIREMENTS:
1. Complex Logic Sections:
 - Must break down multi-step operations
 - Explain interactions between components
 - Detail data flow and transformations
2. Error Handling:
 - Document all error conditions
 - Explain recovery procedures
 - Detail logging and reporting
3. API Interactions:
 - Document all platform method calls
 - Explain data transformations
 - Detail error handling
4. Financial Aid Impact:
 - Explain effects on aid calculations
 - Document regulatory compliance
 - Detail student eligibility impact
Example Section With All Aspects:
// CONDITIONALS:
// Check 1: Payment Period Status Validation
// Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
```

```
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.LetterCodeCriteriaScript)
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
return letterCode.querySql('''
SELECT CASE WHEN :notificationType IN ('TITLE_IV_LOANS_EXIT_COUNSELING_FOR_GRADUATING_STUDENT',
 'TEACH_EXIT_COUNSELING_FOR_GRADUATING_STUDENT',
 'TITLE_IV_LOANS_EXIT_COUNSELING_FOR_WITHDRAWN_STUDENT', 'TEACH_EXIT_COUNSELING_FOR_WITHDRAWN_STUDENT',
 'TITLE_IV_LOANS_EXIT_COUNSELING_FOLLOW_UP_FOR_GRADUATING_STUDENT',
 'TEACH_EXIT_COUNSELING_FOLLOW_UP_FOR_GRADUATING_STUDENT',
 'TITLE_IV_LOANS_EXIT_COUNSELING_FOLLOW_UP_FOR_WITHDRAWN_STUDENT',
 'TEACH_EXIT_COUNSELING_FOLLOW_UP_FOR_WITHDRAWN_STUDENT'
 )
 THEN '200'
 END AS letter_code
FROM DUAL
''')
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Notification Required Criteria

Name: notificationRequiredCriteria

Purpose: Determines if a student notification must be triggered.

In Baseline Config: Yes

Return Type: Boolean

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "notificationRequiredCriteria"
Primary Function: Determines if a student notification must be triggered when an ISIR is matched to a
 student or when the student
package is updated.
Expected Output Type: Boolean
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
acys
Purpose: Provides access to all non-canceled academic years.
Type: IAcademicYearsScheduleAPI
Available Methods:
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
awy
Purpose: The award year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
awyPeriod
Purpose: The period of the award year of the funding element being evaluated.
Type: IAwardYearAPI
Available Methods:
* asList
 - Returns: IAwardYearsAPI
 - Function: Returns an IAwardYearsAPI instance containing only this Award Year.'
* getEndYear
 - Returns: int
 - Function: Gets the end year of this Award Year, for example 2023 for Award Year 2022-2023.'
* getNext
 - Returns: IAwardYearAPI
```

```
 - Function: Gets the Award Year that follows this one chronologically.'
* getPrevious
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that precedes this one chronologically.'
* getStartYear
 - Returns: int
 - Function: Gets the start year of this Award Year, for example 2022 for Award Year 2022-2023.'
* getYear
 - Returns: int
 - Function: Short for getEndYear().'
coa
Purpose: Provides access to all Cost of Attendance information for the student.
Type: ICoaAPI
Available Methods:
* getActiveCoa
 - Returns: int
 - Function: Gets the active cost of attendance.'
* getActualCoa
 - Returns: int
 - Function: Gets the actual cost of attendance.'
* getComponents
 - Returns: ICoaComponentsAPI
 - Function: Gets the COA components.'
* getPellCoa
 - Returns: int
 - Function: Gets the PELL cost of attendance.'
* getTotalCoa
 - Returns: int
 - Function: Gets the total cost of attendance.'
* in
 - Returns: ICoaAPI
 - Function: Filters by term.'
* in
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inAcademicYear
 - Returns: ICoaAPI
 - Function: Filters by academic year.'
* inTerm
 - Returns: ICoaAPI
 - Function: Filters by term.'
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
```

```
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
No methods available
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
```

```
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
dloan
Purpose: Information related to Direct Loans.
Type: IDloanAPI
Available Methods:
* getLifetimeLimit
 - Returns: int
 - Function: No description available'
* getRemainingNsldsLimit
 - Returns: int
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
```

```
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
financialPlan
Purpose: Information about the student's current financial plan.
Type: IFinancialPlanAPI
Available Methods:
* filterCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for CommonLine Unique Ids passing a
 given predicate.'
* getAcyCoa
 - Returns: int
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getAcyUnpackagedAmount
 - Returns: BigDecimal
 - Function: Deprecated.
Consider using the coa binding (ICoaAPI).'
* getCarryOn
 - Returns: BigDecimal
 - Function: No description available'
* getCommonLineUniqueIds
 - Returns: Set<ICommonLineUniqueId>
 - Function: No description available'
* getEnrollmentIntensities
 - Returns: ISet<BigDecimal>
 - Function: No description available'
* getEnrollmentStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getFundStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getMaxAmount
 - Returns: BigDecimal
 - Function: No description available'
* getProratedLimit
 - Returns: BigDecimal
 - Function: No description available'
* getVersion
 - Returns: int
 - Function: Gets the package version.'
* getWithCommonLineUniqueId
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* getWithCommonLineUniqueIds
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of data for the given CommonLine Unique Ids.'
* isEligible
 - Returns: boolean
 - Function: No description available'
* isIsirUsedForCalculation
 - Returns: boolean
 - Function: Identifies ISIRs used while calculating this Financial Plan.'
* isProrated
```

```
 - Returns: boolean
 - Function: No description available'
* isUnmatchedCommonLineLoan
 - Returns: boolean
 - Function: Checks if a given CommonLine Loan (probably coming from the commonLineLoan binding) has not
 been matched
yet in the fund collection.'
fundAcceptance
Purpose: Provides access to all Fund Acceptance information for the student.
Type: IFundAcceptanceAPI
Available Methods:
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for given fund
 acceptance statuses.'
* filterNullableStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for fund acceptance
 statuses passing a given predicate.'
* getNullableStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getValidatedAcceptedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of fund acceptance data for the given fund
 acceptance status.'
fundCodeConfig
Purpose: Provides information about fund code configuration.
Type: IFundCodeConfigAPI
Available Methods:
* getFundCodesByPriority
 - Returns: List<String>
 - Function: No description available'
* getHigherPriorityFundCodes
 - Returns: List<String>
 - Function: No description available'
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
```

```
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isirRecord
Purpose: Matched ISIR that triggered the script evaluation when the event trigger is "ISIR_MATCHED", or the
 active ISIR
with the highest award year in the updated package when the event trigger is "PACKAGE_UPDATED".
Type: IIsirRecordAPI
Available Methods:
* getActiveIsirForAwardYear
 - Returns: Boolean
 - Function: Gets the active ISIR for award year status.'
* getAllFields
 - Returns: @NonNull Map<IsirField<?>,IIsirFieldValue<?>>
 - Function: Retrieves all field values (WARNING: can only be used with 2025+ ISIRs).'
* getAssumptions
 - Returns: List<IAssumptionAPI>
 - Function: Get the assumptions.'
* getAwardYear
 - Returns: String
 - Function: Gets the award year.'
* getCommentCodes
 - Returns: List<ICommentCodeAPI>
 - Function: Get the comment codes.'
* getCpsTransNumber
 - Returns: String
 - Function: Gets the CPS transaction number.'
* getCurrentSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's current SSN.'
* getDiscrepancies
 - Returns: List<IDiscrepancyAPI>
 - Function: Get the discrepancies.'
* getField
 - Returns: <T> @NonNull IIsirFieldValue<T>
 - Function: Retrieves the value of a given field (WARNING: can only be used with 2025+ ISIRs).'
* getFirstName
 - Returns: String
 - Function: Gets the student's first name.'
* getHighlightFlags
 - Returns: List<IHighlightFlagAPI>
 - Function: Get the highlight flags.'
* getIsirFieldValue
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValueOrNull
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValues
 - Returns: Map<String,String>
 - Function: Get a map containing all ISIR fields (WARNING: This method can only be used up to 2024 award
 year).'
```

```
* getIsirIdOriginalNameId
 - Returns: String
 - Function: Gets the student's original name ID.'
* getIsirIdSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's original SSN.'
* getIsirIdYearIndicator
 - Returns: String
 - Function: Gets the ISIR's year indicator.'
* getIsirStatus
 - Returns: String
 - Function: Gets the ISIR status.'
* getIsirStatusBeforeCorrection
 - Returns: String
 - Function: When ISIR Status is Corrected returns the previous ISIR Status.'
* getLastName
 - Returns: String
 - Function: Gets the student's last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the student's middle name.'
* getReceivedDate
 - Returns: Date
 - Function: Gets the system receipt date of the ISIR.'
* getRejectCodes
 - Returns: List<IRejectCodeAPI>
 - Function: Get the reject codes.'
* getSchoolCode1
 - Returns: String
 - Function: Gets the school code 1.'
* getSchoolCode10
 - Returns: String
 - Function: Gets the school code 10.'
* getSchoolCode2
 - Returns: String
 - Function: Gets the school code 2.'
* getSchoolCode3
 - Returns: String
 - Function: Gets the school code 3.'
* getSchoolCode4
 - Returns: String
 - Function: Gets the school code 4.'
* getSchoolCode5
 - Returns: String
 - Function: Gets the school code 5.'
* getSchoolCode6
 - Returns: String
 - Function: Gets the school code 6.'
* getSchoolCode7
 - Returns: String
 - Function: Gets the school code 7.'
* getSchoolCode8
 - Returns: String
 - Function: Gets the school code 8.'
* getSchoolCode9
 - Returns: String
 - Function: Gets the school code 9.'
* isCompatibleWithPackaging
 - Returns: boolean
 - Function: Checks if this ISIR satisfies minimal acceptance criteria for automatic Financial
Aid Package Calculation.'
* isUnverified
 - Returns: boolean
 - Function: Gets the ISIR verification status.'
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
```

```
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
```

```
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
```

```
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
```

```
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
```

```
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
```

```
Available Methods:
* getPeriodsIntersectingWithGroupForThePeriod
 - Returns: List<? extends IPeriod>
 - Function: Find the periods student was in the given group in the specified period of time'
* isInGroupAtAnyPointInThePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period at any point of time.'
* isInGroupAtDate
 - Returns: boolean
 - Function: Checks if student was in the given group on the given date.'
* isInGroupDuringCompletePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period for the whole time.'
* numberOfDaysInGroupForThePeriod
 - Returns: int
 - Function: Count number of days student was in the given group in the specified period of time'
* getActiveGroupsAtAnyPointInThePeriod
 - Returns: List<String>
 - Function: Find the student's groups active at any point during specified period of time'
* getActiveGroupsAtDate
 - Returns: List<String>
 - Function: Find the student's active groups at the given date'
* getActiveGroupsDuringCompletePeriod
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
```

```
    // Check [number]: [specific limit being enforced]
    // Source: [where boundary values come from]
    // Variables: [variables and methods involved]
    // Validation:
    // - [how boundaries are checked]
    // - [validation method details]
    // Edge Cases:
    // - [specific edge cases handled]
    // - [how each is managed]
    // Error Handling: [boundary violation management]
    // Impact: [effect of boundary violations]
    // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
  3. ARITHMETIC Section:
    Format:
    // ARITHMETIC:
    // Operation [number]: [name of calculation]
    // Formula: [complete mathematical expression]
    // Variables:
    // - [each variable and its purpose]
    // - [data types and units]
    // Steps:
    // 1. [detailed step-by-step breakdown]
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
  ADDITIONAL REQUIREMENTS:
  1. Complex Logic Sections:
    - Must break down multi-step operations
    - Explain interactions between components
    - Detail data flow and transformations
  2. Error Handling:
    - Document all error conditions
    - Explain recovery procedures
    - Detail logging and reporting
  3. API Interactions:
    - Document all platform method calls
    - Explain data transformations
    - Detail error handling
  4. Financial Aid Impact:
    - Explain effects on aid calculations
    - Document regulatory compliance
    - Detail student eligibility impact
  Example Section With All Aspects:
  // CONDITIONALS:
  // Check 1: Payment Period Status Validation
  // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
  // Purpose: Ensure only valid, active payment periods are included
  // Logic:
  // - Retrieve all loan payment periods
  // - Filter based on status (!= "CANCELED")
  // - Compare start dates with reference date
  // Success Path: Period included in calculations
  // Failure Path: Period excluded from totals
  // Business Rules: Canceled periods must not affect grade level
  // Error Handling: Invalid periods logged and skipped
  // Financial Impact: Affects available loan amounts
  //
  // BOUNDARY CHECKS:
  // Check 1: Payment Period Date Validation
  // Source: Academic Calendar Configuration
  // Variables: period.getStartDate(), referenceDate
  // Validation:
```

```
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.NotificationRequiredCriteriaScript)
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
return primaryProgram.getEnrollmentStatus() != null && financialPlan.getFunds(["GPLUS"]);
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# COA Cost Value

Name: coaCostValue

Purpose: Calculates the value of a COA item.

In Baseline Config: Yes

Return Type: BigDecimal

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
```

```
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "coaCostValue"
Primary Function: Calculates the value of a COA item.
Expected Output Type: BigDecimal
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
scope
Purpose: Gathers all the scope details of the script being evaluated (acyNo, fundCode,
fundType, awy, lpNo, ppNo) allowing quick access to financialPlan/disbursements/etc
subsets directly pertaining to this scope.
Type: IFundScopeAPI
Available Methods:
* getAcyNo
 - Returns: int
 - Function: No description available'
* getAwy
 - Returns: int
 - Function: No description available'
* getFundCode
 - Returns: String
 - Function: No description available'
* getFundType
 - Returns: String
 - Function: No description available'
* getLpNo
 - Returns: int
 - Function: No description available'
* getPpNo
 - Returns: int
 - Function: No description available'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
```

```
    - Function: Gets the dependency status.'
  * getDeterminationDate
    - Returns: Date
    - Function: Gets the determination date.'
  * getDocuments
    - Returns: Map<String,IReceivedDoc>
    - Function: Gets the student's documents, indexed by document code.'
  * getEmail
    - Returns: String
    - Function: Gets the e-mail.'
  * getEnrollmentStatus
    - Returns: String
    - Function: Gets the enrollment status.'
  * getExternalStudentId
    - Returns: String
    - Function: Gets the external student ID (unique per student).'
  * getFinancialAidRequested
    - Returns: Boolean
    - Function: Gets the financial aid request status.'
  * getFirstName
    - Returns: String
    - Function: Gets the first name.'
  * getGender
    - Returns: String
    - Function: Gets the gender.'
  * getHighSchoolEquivalency
    - Returns: Boolean
    - Function: Gets the high school equivalency status.'
  * getLastName
    - Returns: String
    - Function: Gets the last name.'
  * getMiddleName
    - Returns: String
    - Function: Gets the middle name.'
  * getPersonId
    - Returns: Long
    - Function: Gets the person ID.'
  * getProgramType
    - Returns: String
    - Function: Gets the program type.'
  * getRecentFinancialAidHistoryRecords
    - Returns: IRecentFahRecordsAPI
    - Function: Gets the recent financial aid records.'
  * getSchoolCode
    - Returns: String
    - Function: Gets the school code.'
  * getSchoolEmployee
    - Returns: Boolean
    - Function: Gets the school employee status.'
  * getSchoolEmployeeDependent
    - Returns: Boolean
    - Function: Gets the school employee dependent status.'
  * getSsn
    - Returns: ISocialSecurityNumber
    - Function: Gets the SSN.'
  * getStudentId
    - Returns: Long
    - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
Available Methods:
  * getPeriodsIntersectingWithGroupForThePeriod
    - Returns: List<? extends IPeriod>
    - Function: Find the periods student was in the given group in the specified period of time'
  * isInGroupAtAnyPointInThePeriod
```

```
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period at any point of time.'
* isInGroupAtDate
 - Returns: boolean
 - Function: Checks if student was in the given group on the given date.'
* isInGroupDuringCompletePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period for the whole time.'
* numberOfDaysInGroupForThePeriod
 - Returns: int
 - Function: Count number of days student was in the given group in the specified period of time'
* getActiveGroupsAtAnyPointInThePeriod
 - Returns: List<String>
 - Function: Find the student's groups active at any point during specified period of time'
* getActiveGroupsAtDate
 - Returns: List<String>
 - Function: Find the student's active groups at the given date'
* getActiveGroupsDuringCompletePeriod
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
term
Purpose: A period matching the term being evaluated.
Type: IPeriod
Available Methods:
No methods available
termNo
Purpose: The number of the term of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
acy
Purpose: The academic year of the funding element being evaluated.
Type: IAcademicYearScheduleAPI
Available Methods:
* getCompletedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getGradeLevel
 - Returns: int
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getPeriodOfEnrollmentEndDate
 - Returns: LocalDate
 - Function: No description available'
* getPeriodOfEnrollmentStartDate
 - Returns: LocalDate
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getUnits
 - Returns: BigDecimal
```

```
 - Function: No description available'
acyNo
Purpose: The number of the academic year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
acyOrder
Purpose: The order number of the academic year of the funding element being evaluated, among all non-
canceled academic
years.
Type: Integer
Available Methods:
No methods available
acys
Purpose: Provides access to all non-canceled academic years.
Type: IAcademicYearsScheduleAPI
Available Methods:
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
awy
Purpose: The award year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
awyPeriod
Purpose: The period of the award year of the funding element being evaluated.
Type: IAwardYearAPI
Available Methods:
* asList
 - Returns: IAwardYearsAPI
 - Function: Returns an IAwardYearsAPI instance containing only this Award Year.'
* getEndYear
 - Returns: int
 - Function: Gets the end year of this Award Year, for example 2023 for Award Year 2022-2023.'
* getNext
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that follows this one chronologically.'
* getPrevious
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that precedes this one chronologically.'
* getStartYear
 - Returns: int
 - Function: Gets the start year of this Award Year, for example 2022 for Award Year 2022-2023.'
* getYear
 - Returns: int
 - Function: Short for getEndYear().'
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
```

```
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
No methods available
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
```

```
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
```

```
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
fundCode
Purpose: The fund code of the funding element being evaluated.
Type: String
Available Methods:
No methods available
fundType
Purpose: The fund type of the funding element being evaluated.
Type: String
Available Methods:
No methods available
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
```

```
 - Function: Returns a IPeriod given its start and end dates.'
isFirstAcy
Purpose: True if the academic year of the funding element being evaluated is the first one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isLastAcy
Purpose: True if the academic year of the funding element being evaluated is the last one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
lp
Purpose: The loan period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
lpNo
Purpose: The number of the loan period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
pp
Purpose: The payment period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
```

```
 - Returns: BigDecimal
 - Function: No description available'
ppNo
Purpose: The number of the payment period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
```

```
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
```

```
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
```

```
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
```

```
    // Check [number]: [specific limit being enforced]
    // Source: [where boundary values come from]
    // Variables: [variables and methods involved]
    // Validation:
    // - [how boundaries are checked]
    // - [validation method details]
    // Edge Cases:
    // - [specific edge cases handled]
    // - [how each is managed]
    // Error Handling: [boundary violation management]
    // Impact: [effect of boundary violations]
    // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
  3. ARITHMETIC Section:
    Format:
    // ARITHMETIC:
    // Operation [number]: [name of calculation]
    // Formula: [complete mathematical expression]
    // Variables:
    // - [each variable and its purpose]
    // - [data types and units]
    // Steps:
    // 1. [detailed step-by-step breakdown]
    // 2. [intermediate calculations]
    // 3. [transformations applied]
    // Business Context: [why this calculation is needed]
    // Result: [meaning of the output]
    // Financial Impact: [effect on aid calculations]
    // Limitations: [any mathematical constraints]
    // If N/A, explicitly state: "ARITHMETIC: N/A"
  ADDITIONAL REQUIREMENTS:
  1. Complex Logic Sections:
    - Must break down multi-step operations
    - Explain interactions between components
    - Detail data flow and transformations
  2. Error Handling:
    - Document all error conditions
    - Explain recovery procedures
    - Detail logging and reporting
  3. API Interactions:
    - Document all platform method calls
    - Explain data transformations
    - Detail error handling
  4. Financial Aid Impact:
    - Explain effects on aid calculations
    - Document regulatory compliance
    - Detail student eligibility impact
  Example Section With All Aspects:
  // CONDITIONALS:
  // Check 1: Payment Period Status Validation
  // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
  // Purpose: Ensure only valid, active payment periods are included
  // Logic:
  // - Retrieve all loan payment periods
  // - Filter based on status (!= "CANCELED")
  // - Compare start dates with reference date
  // Success Path: Period included in calculations
  // Failure Path: Period excluded from totals
  // Business Rules: Canceled periods must not affect grade level
  // Error Handling: Invalid periods logged and skipped
  // Financial Impact: Affects available loan amounts
  //
  // BOUNDARY CHECKS:
  // Check 1: Payment Period Date Validation
  // Source: Academic Calendar Configuration
  // Variables: period.getStartDate(), referenceDate
  // Validation:
```

```
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.CoaCostValueScript)
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
import org.joda.time.LocalDate
def referencePeriod = program.isTerm() ? term : acy
// Institutions should define their own add drop policy
LocalDate addDropPolicy = referencePeriod.getStartDate().plusDays(10);
def countedUnits = 0.0;
String programType = primaryProgram.getProgramType();
// If the Program Enrollment Status is not X, then count credits
if (primaryProgram.getEnrollmentStatus() != null && primaryProgram.getEnrollmentStatus() != "X")
{
 for (def course : program.getCourses().getAssociatedTo(referencePeriod))
 {
 log.debug("COA SCRIPT Tuition, COURSE IN LOOP: countedUnits = {}, courseEndDate = {}, courseAraIndicator
 = {}, courseSchedulingStatus = {}, courseUnitsToAdd = {}", countedUnits, course.getEndDate(),
 course.getFirstAraIndicator(), course.getSchedulingStatus(), course.getUnits());
 // If the course is withdrawn, only count the credits if the course was withdrawn after the add/drop
 period. If the course is not withdrawn, count the course credits.
 if ((["WITHDRAWN"].contains(course.getSchedulingStatus()) && course.getLastDateOfAttendance() != null &&
 course.getLastDateOfAttendance() > addDropPolicy) || course.getSchedulingStatus() != "WITHDRAWN")
 {
 countedUnits = countedUnits + course.getUnits();
 }
 }
}
log.debug("COA SCRIPT Tuition - final value of countedUnits = {}", countedUnits);
// Use different cost values for different program types
if (programType == "ASSOCIATE" || programType == "BACHELORS")
{
 return (countedUnits * 175);
}
else
{
 return (countedUnits * 225);
}
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
```

```
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Additional COA Components

Name: additionalCoaComponents

Purpose: Evaluates if a COA configuration is applicable.

In Baseline Config: No

Return Type: Boolean

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "additionalCoaComponents"
Primary Function: Evaluates if a COA configuration is applicable to a student.
Expected Output Type: Boolean
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
acy
Purpose: The academic year of the funding element being evaluated.
Type: IAcademicYearScheduleAPI
Available Methods:
* getCompletedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getGradeLevel
 - Returns: int
 - Function: No description available'
```

```
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getMonths
 - Returns: BigDecimal
 - Function: No description available'
* getMonthsFromPeriodDates
 - Returns: int
 - Function: No description available'
* getPeriodOfEnrollmentEndDate
 - Returns: LocalDate
 - Function: No description available'
* getPeriodOfEnrollmentStartDate
 - Returns: LocalDate
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
acyNo
Purpose: The number of the academic year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
acyOrder
Purpose: The order number of the academic year of the funding element being evaluated, among all non-
canceled academic
years.
Type: Integer
Available Methods:
No methods available
acys
Purpose: Provides access to all non-canceled academic years.
Type: IAcademicYearsScheduleAPI
Available Methods:
* getLoanPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
* getTerms
 - Returns: IAcademicTermPeriodsAPI
 - Function: No description available'
* getGrantPaymentPeriods
 - Returns: ISimpleNumberedPeriods<ISimpleScheduledPeriodAPI>
 - Function: No description available'
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
awy
Purpose: The award year of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
awyPeriod
Purpose: The period of the award year of the funding element being evaluated.
Type: IAwardYearAPI
Available Methods:
* asList
 - Returns: IAwardYearsAPI
 - Function: Returns an IAwardYearsAPI instance containing only this Award Year.'
```

```
* getEndYear
 - Returns: int
 - Function: Gets the end year of this Award Year, for example 2023 for Award Year 2022-2023.'
* getNext
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that follows this one chronologically.'
* getPrevious
 - Returns: IAwardYearAPI
 - Function: Gets the Award Year that precedes this one chronologically.'
* getStartYear
 - Returns: int
 - Function: Gets the start year of this Award Year, for example 2022 for Award Year 2022-2023.'
* getYear
 - Returns: int
 - Function: Short for getEndYear().'
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
No methods available
disbursements
Purpose: Information about all student disbursements.
Type: IDisbursementsAPI
Available Methods:
* filterSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for sequence numbers
 passing a given predicate.'
* filterStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursement statuses
 passing a given predicate.'
* getCancelled
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements that
 have been cancelled.'
* getDisbursed
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements
that have not been cancelled (see getCancelled(boolean), include disbursed payments
and whose individual Total Disbursement Amounts are positive.'
```

```
* getIncrementalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLateDisbursement
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given Late Disbursement
Indicator.'
* getPpDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getLatestPayments
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data with only latest
 payments.'
* getLockingPp
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for (non-)payment period-
locking disbursements'
* getPpAppliedAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPwd
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for disbursements with the
 given PWD Indicator.'
* getSeqNos
 - Returns: Set<Integer>
 - Function: No description available'
* getStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getTotalDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getWithSeqNo
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for a given sequence
 number.'
* getWithSeqNos
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given sequence
 numbers.'
* getWithStatus
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for the given disbursement
 status.'
* getWithStatuses
 - Returns: T
 - Function: Creates a narrowed-down, independent collection of disbursement data for given disbursement
 statuses.'
* hasBeenCancelled
 - Returns: boolean
 - Function: No description available'
* isDisbursed
 - Returns: boolean
 - Function: No description available'
* isLateDisbursement
 - Returns: boolean
 - Function: No description available'
* isLockingPp
 - Returns: boolean
 - Function: No description available'
* isPwd
 - Returns: boolean
 - Function: No description available'
```

```
* getPpMaxDisbursementAmount
 - Returns: BigDecimal
 - Function: No description available'
* getPpTotalDecreasedAmount
 - Returns: BigDecimal
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
fundCode
Purpose: The fund code of the funding element being evaluated.
Type: String
Available Methods:
No methods available
fundType
Purpose: The fund type of the funding element being evaluated.
Type: String
Available Methods:
No methods available
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
```

```
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
```

```
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
Available Methods:
* getPeriodsIntersectingWithGroupForThePeriod
 - Returns: List<? extends IPeriod>
 - Function: Find the periods student was in the given group in the specified period of time'
* isInGroupAtAnyPointInThePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period at any point of time.'
* isInGroupAtDate
 - Returns: boolean
 - Function: Checks if student was in the given group on the given date.'
* isInGroupDuringCompletePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period for the whole time.'
* numberOfDaysInGroupForThePeriod
 - Returns: int
 - Function: Count number of days student was in the given group in the specified period of time'
* getActiveGroupsAtAnyPointInThePeriod
 - Returns: List<String>
 - Function: Find the student's groups active at any point during specified period of time'
* getActiveGroupsAtDate
 - Returns: List<String>
 - Function: Find the student's active groups at the given date'
* getActiveGroupsDuringCompletePeriod
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
term
Purpose: A period matching the term being evaluated.
Type: IPeriod
Available Methods:
No methods available
termNo
Purpose: The number of the term of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
```

```
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isFirstAcy
Purpose: True if the academic year of the funding element being evaluated is the first one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isLastAcy
Purpose: True if the academic year of the funding element being evaluated is the last one among all non-
canceled academic
years.
Type: Boolean
Available Methods:
No methods available
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
lp
Purpose: The loan period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
lpNo
Purpose: The number of the loan period of the funding element being evaluated.
```

```
Type: Integer
Available Methods:
No methods available
pp
Purpose: The payment period of the funding element being evaluated.
Type: ISimpleScheduledPeriodAPI
Available Methods:
* getUnits
 - Returns: BigDecimal
 - Function: No description available'
ppNo
Purpose: The number of the payment period of the funding element being evaluated.
Type: Integer
Available Methods:
No methods available
previousFinancialPlan
Purpose: Provides access to previous financial plans.
Type: IPreviousFinancialPlanAPI
Available Methods:
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given time.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective on a given local date.'
* effectiveAt
 - Returns: T
 - Function: Gets a previous financial plan that was effective at a given local time.'
* withVersion
 - Returns: T
 - Function: Gets a previous financial plan with a given version.'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
```

```
    - Returns: ICoursesAPI
    - Function: No description available'
  * getDateOfDetermination
    - Returns: LocalDate
    - Function: No description available'
  * getDescription
    - Returns: String
    - Function: No description available'
  * getEnrollmentStatus
    - Returns: String
    - Function: No description available'
  * getEnrollmentStatusEffectiveDate
    - Returns: LocalDate
    - Function: No description available'
  * getEnrollmentStatusSubType
    - Returns: String
    - Function: No description available'
  * getExternalProgramId
    - Returns: String
    - Function: No description available'
  * getFaEligibleIndicator
    - Returns: Boolean
    - Function: No description available'
  * getGpa
    - Returns: BigDecimal
    - Function: No description available'
  * getManualSapEvaluationIndicator
    - Returns: Boolean
    - Function: No description available'
  * getModality
    - Returns: String
    - Function: No description available'
  * getOfficialLastDateOfAttendance
    - Returns: LocalDate
    - Function: No description available'
  * getOpeId
    - Returns: String
    - Function: No description available'
  * getPrimaryLocation
    - Returns: String
    - Function: No description available'
  * getProgramLengthMeasurementUnit
    - Returns: String
    - Function: No description available'
  * getProgramLengthNumberOfUnits
    - Returns: int
    - Function: No description available'
  * getProgramMeasurementUnit
    - Returns: String
    - Function: No description available'
  * getProgramType
    - Returns: String
    - Function: No description available'
  * getRequiredGpa
    - Returns: BigDecimal
    - Function: No description available'
  * getSpecialPrograms
    - Returns: String
    - Function: No description available'
  * getStartDate
    - Returns: LocalDate
    - Function: No description available'
  * getStudentResources
    - Returns: List<IStudentResourceAPI>
    - Function: No description available'
  * getTermAcys
```

```
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
r2t4
Purpose: Provides access to all R2T4 information for the student.
Type: IR2t4API
Available Methods:
* filterOldas
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for OLDAs passing a given
 predicate.'
* filterProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for process statuses passing a
 given predicate.'
* getGrossAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNetAmountDecreased
 - Returns: BigDecimal
 - Function: No description available'
* getNullableOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getNullableProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getOldas
 - Returns: ISet<LocalDate>
 - Function: No description available'
* getProcessStatuses
 - Returns: ISet<String>
 - Function: No description available'
* getWithLatest
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given isLatest status.'
* getWithOlda
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process OLDA.'
* getWithOldas
```

```
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process OLDAs.'
* getWithProcessStatus
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for the given process status.'
* getWithProcessStatuses
 - Returns: IR2t4API
 - Function: Creates a narrowed-down, independent collection of R2T4 data for given process statuses.'
* isEligibleForPWD
 - Returns: boolean
 - Function: No description available'
* isLatest
 - Returns: boolean
 - Function: Checks if at least one R2T4 process in the collection is the latest.'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
scope
Purpose: Gathers all the scope details of the script being evaluated (acyNo, fundCode,
fundType, awy, lpNo, ppNo) allowing quick access to financialPlan/disbursements/etc
subsets directly pertaining to this scope.
Type: IFundScopeAPI
Available Methods:
* getAcyNo
 - Returns: int
 - Function: No description available'
* getAwy
 - Returns: int
 - Function: No description available'
* getFundCode
 - Returns: String
 - Function: No description available'
* getFundType
 - Returns: String
 - Function: No description available'
* getLpNo
 - Returns: int
 - Function: No description available'
* getPpNo
 - Returns: int
 - Function: No description available'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
```

```
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
 // Operation [number]: [name of calculation]
 // Formula: [complete mathematical expression]
 // Variables:
 // - [each variable and its purpose]
 // - [data types and units]
 // Steps:
 // 1. [detailed step-by-step breakdown]
 // 2. [intermediate calculations]
 // 3. [transformations applied]
 // Business Context: [why this calculation is needed]
 // Result: [meaning of the output]
 // Financial Impact: [effect on aid calculations]
 // Limitations: [any mathematical constraints]
 // If N/A, explicitly state: "ARITHMETIC: N/A"
ADDITIONAL REQUIREMENTS:
1. Complex Logic Sections:
 - Must break down multi-step operations
 - Explain interactions between components
 - Detail data flow and transformations
```

```
2. Error Handling:
 - Document all error conditions
 - Explain recovery procedures
 - Detail logging and reporting
3. API Interactions:
 - Document all platform method calls
 - Explain data transformations
 - Detail error handling
4. Financial Aid Impact:
 - Explain effects on aid calculations
 - Document regulatory compliance
 - Detail student eligibility impact
Example Section With All Aspects:
// CONDITIONALS:
// Check 1: Payment Period Status Validation
// Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
N/A - There is no baseline config. This is a placeholder.
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
```

```
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Exit Counseling Graduating Criteria

Name: exitCounselingGraduatingCriteria

Purpose: Evaluates the exit counseling criteria for graduating students.

In Baseline Config: Yes

Return Type: Boolean

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "exitCounselingGraduatingCriteria"
Primary Function: Evaluates exit counseling criteria for graduating students.
Expected Output Type: Boolean
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
```

```
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
No methods available
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
```

```
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
```

```
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
```

```
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
receivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
```

```
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
```

```
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
Available Methods:
* getPeriodsIntersectingWithGroupForThePeriod
 - Returns: List<? extends IPeriod>
 - Function: Find the periods student was in the given group in the specified period of time'
* isInGroupAtAnyPointInThePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period at any point of time.'
* isInGroupAtDate
 - Returns: boolean
 - Function: Checks if student was in the given group on the given date.'
* isInGroupDuringCompletePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period for the whole time.'
* numberOfDaysInGroupForThePeriod
 - Returns: int
 - Function: Count number of days student was in the given group in the specified period of time'
* getActiveGroupsAtAnyPointInThePeriod
 - Returns: List<String>
 - Function: Find the student's groups active at any point during specified period of time'
* getActiveGroupsAtDate
 - Returns: List<String>
 - Function: Find the student's active groups at the given date'
* getActiveGroupsDuringCompletePeriod
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
```

```
   - Financial aid implications
   - Required inputs and expected outputs
   - Error handling approach
   - Integration points
   - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
   Format:
   // CONDITIONALS:
   // Check [number]: [specific condition being evaluated]
   // Variables: [list of variables and methods involved]
   // Purpose: [detailed business reason for this check]
   // Logic:
   // - [step-by-step breakdown of the condition]
   // - [how variables are used]
   // - [any transformations applied]
   // Success Path: [what happens when condition is true]
   // Failure Path: [what happens when condition is false]
   // Business Rules: [specific rules being enforced]
   // Error Handling: [how errors are managed]
   // Financial Impact: [effect on aid calculations]
   // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
   Format:
   // BOUNDARY CHECKS:
   // Check [number]: [specific limit being enforced]
   // Source: [where boundary values come from]
   // Variables: [variables and methods involved]
   // Validation:
   // - [how boundaries are checked]
   // - [validation method details]
   // Edge Cases:
   // - [specific edge cases handled]
   // - [how each is managed]
   // Error Handling: [boundary violation management]
   // Impact: [effect of boundary violations]
   // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
   Format:
   // ARITHMETIC:
   // Operation [number]: [name of calculation]
   // Formula: [complete mathematical expression]
   // Variables:
   // - [each variable and its purpose]
   // - [data types and units]
   // Steps:
   // 1. [detailed step-by-step breakdown]
   // 2. [intermediate calculations]
   // 3. [transformations applied]
   // Business Context: [why this calculation is needed]
   // Result: [meaning of the output]
   // Financial Impact: [effect on aid calculations]
   // Limitations: [any mathematical constraints]
   // If N/A, explicitly state: "ARITHMETIC: N/A"
ADDITIONAL REQUIREMENTS:
1. Complex Logic Sections:
   - Must break down multi-step operations
   - Explain interactions between components
   - Detail data flow and transformations
2. Error Handling:
   - Document all error conditions
   - Explain recovery procedures
   - Detail logging and reporting
3. API Interactions:
   - Document all platform method calls
```

```
 - Explain data transformations
 - Detail error handling
4. Financial Aid Impact:
 - Explain effects on aid calculations
 - Document regulatory compliance
 - Detail student eligibility impact
Example Section With All Aspects:
// CONDITIONALS:
// Check 1: Payment Period Status Validation
// Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
//file:noinspection UnnecessaryQualifiedReference
@groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.ExitCounselingGraduatingCriteriaScript)
package CHANGE_ME // this should be updated to your actual package name
import com.oracle.sfp.scripting.api.*
import com.oracle.sfp.scripting.api.util.*
import org.joda.time.LocalDate;
boolean returnValue = false;
LocalDate academicCompletionDate = primaryProgram.getAcademicCompletionDate();
/* no API available for disbursements, and no way to separate out TEACH criteria based on current design */
if (academicCompletionDate != null)
{
 returnValue = true;
}
return returnValue;
```
## COMMENT GENERATION REQUIREMENTS:
```

```
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Exit Counseling Withdrawn Criteria

Name: exitCounselingWithdrawnCriteria

Purpose: Evaluates the exit counseling criteria for withdrawn students.

In Baseline Config:

Return Type: Boolean

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "exitCounselingWithdrawnCriteria"
Primary Function: Evaluates exit counseling criteria for withdrawn students.
Expected Output Type: Boolean
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
allAwardYearIsirs
Purpose: Provides access to all ISIRs for the student, accessible per award year.
Type: IAllAwardYearIsirsAPI
Available Methods:
No methods available
cod
Purpose: Information received from COD for the student.
Type: ICodAPI
Available Methods:
* getCracs
 - Returns: IList<ICounselingCodInfoAPI>
```

```
 - Function: No description available'
* getCrats
 - Returns: IList<ICratInfoAPI>
 - Function: No description available'
* getCrcos
 - Returns: IList<ICrcoInfoAPI>
 - Function: No description available'
* getCrcses
 - Returns: IList<ICrcsInfoAPI>
 - Function: No description available'
* getCrecs
 - Returns: IList<ICounselingCodInfoAPI>
 - Function: No description available'
* getCrpns
 - Returns: IList<ICrpnInfoAPI>
 - Function: No description available'
* getCrsps
 - Returns: IList<ICrspInfoAPI>
 - Function: No description available'
commonLineLoans
Purpose: Collection of CommonLine Loans currently matched to the student.
Type: ICommonLineLoansAPI
Available Methods:
No methods available
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
fafsaAwardYearInfoList
Purpose: Binding that allows scripts to access the FAFSA regulatory dates configuration.
Type: IFafsaAwardYearInfoListAPI
Available Methods:
* distinct
 - Returns: IFafsaAwardYearInfoListAPI
 - Function: Creates a copy of this object by removing repeated occurrences of elements.'
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
```

```
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isirRecords
Purpose: Provides access to the current ISIRs - the ones with the highest CPS number - in each award year.
Type: IIsirsAPI
Available Methods:
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
* getCurrentIsirForAwardYear
 - Returns: IIsirRecordAPI
 - Function: Gets the current ISIR (see explanation in the header to this interface) for a given Award
Year, if present.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
```

```
program
Purpose: Information about the student's primary program.
Type: IProgramAPI
Available Methods:
* exists
 - Returns: boolean
 - Function: No description available'
* getAcademicCompletionDate
 - Returns: LocalDate
 - Function: No description available'
* getAcademicStatus
 - Returns: String
 - Function: No description available'
* getAdmissionStatus
 - Returns: String
 - Function: No description available'
* getAssessedUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydUnits
 - Returns: BigDecimal
 - Function: No description available'
* getAydWeeks
 - Returns: Integer
 - Function: No description available'
* getCharges
 - Returns: List<IChargeAPI>
 - Function: No description available'
* getCode
 - Returns: String
 - Function: No description available'
* getCollege
 - Returns: String
 - Function: No description available'
* getCourses
 - Returns: ICoursesAPI
 - Function: No description available'
* getDateOfDetermination
 - Returns: LocalDate
 - Function: No description available'
* getDescription
 - Returns: String
 - Function: No description available'
* getEnrollmentStatus
 - Returns: String
 - Function: No description available'
* getEnrollmentStatusEffectiveDate
 - Returns: LocalDate
 - Function: No description available'
* getEnrollmentStatusSubType
 - Returns: String
 - Function: No description available'
* getExternalProgramId
 - Returns: String
 - Function: No description available'
* getFaEligibleIndicator
 - Returns: Boolean
 - Function: No description available'
* getGpa
 - Returns: BigDecimal
 - Function: No description available'
* getManualSapEvaluationIndicator
 - Returns: Boolean
 - Function: No description available'
* getModality
 - Returns: String
```

```
 - Function: No description available'
* getOfficialLastDateOfAttendance
 - Returns: LocalDate
 - Function: No description available'
* getOpeId
 - Returns: String
 - Function: No description available'
* getPrimaryLocation
 - Returns: String
 - Function: No description available'
* getProgramLengthMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramLengthNumberOfUnits
 - Returns: int
 - Function: No description available'
* getProgramMeasurementUnit
 - Returns: String
 - Function: No description available'
* getProgramType
 - Returns: String
 - Function: No description available'
* getRequiredGpa
 - Returns: BigDecimal
 - Function: No description available'
* getSpecialPrograms
 - Returns: String
 - Function: No description available'
* getStartDate
 - Returns: LocalDate
 - Function: No description available'
* getStudentResources
 - Returns: List<IStudentResourceAPI>
 - Function: No description available'
* getTermAcys
 - Returns: ISimpleNumberedPeriods<ITermAcademicYearAPI>
 - Function: No description available'
* getTerms
 - Returns: ITermPeriodsAPI
 - Function: No description available'
* getTermType
 - Returns: String
 - Function: No description available'
* getTotalAcceptedTransferUnits
 - Returns: BigDecimal
 - Function: No description available'
* getTotalRequiredUnits
 - Returns: BigDecimal
 - Function: No description available'
* isBbayTerm
 - Returns: boolean
 - Function: No description available'
* isGraduate
 - Returns: boolean
 - Function: No description available'
* isProgramTerm
 - Returns: boolean
 - Function: No description available'
* isTerm
 - Returns: boolean
 - Function: No description available'
* isUndergraduate
 - Returns: boolean
 - Function: No description available'
receivedDocuments
Purpose: Provides access to all documents received for the student.
```

```
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
sap
Purpose: Information about the student's latest SAP.
Type: ISapAPI
Available Methods:
* getLatestStatus
 - Returns: String
 - Function: No description available'
* getLatestStatus
 - Returns: String
 - Function: Gets the status for a given period for the student.'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
 - Returns: String
 - Function: Gets the e-mail.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getExternalStudentId
 - Returns: String
 - Function: Gets the external student ID (unique per student).'
* getFinancialAidRequested
 - Returns: Boolean
 - Function: Gets the financial aid request status.'
* getFirstName
 - Returns: String
 - Function: Gets the first name.'
* getGender
 - Returns: String
 - Function: Gets the gender.'
* getHighSchoolEquivalency
 - Returns: Boolean
 - Function: Gets the high school equivalency status.'
```

```
* getLastName
 - Returns: String
 - Function: Gets the last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the middle name.'
* getPersonId
 - Returns: Long
 - Function: Gets the person ID.'
* getProgramType
 - Returns: String
 - Function: Gets the program type.'
* getRecentFinancialAidHistoryRecords
 - Returns: IRecentFahRecordsAPI
 - Function: Gets the recent financial aid records.'
* getSchoolCode
 - Returns: String
 - Function: Gets the school code.'
* getSchoolEmployee
 - Returns: Boolean
 - Function: Gets the school employee status.'
* getSchoolEmployeeDependent
 - Returns: Boolean
 - Function: Gets the school employee dependent status.'
* getSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the SSN.'
* getStudentId
 - Returns: Long
 - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
studentGroups
Purpose: Student Group information.
Type: IStudentGroupsAPI
Available Methods:
* getPeriodsIntersectingWithGroupForThePeriod
 - Returns: List<? extends IPeriod>
 - Function: Find the periods student was in the given group in the specified period of time'
* isInGroupAtAnyPointInThePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period at any point of time.'
* isInGroupAtDate
 - Returns: boolean
 - Function: Checks if student was in the given group on the given date.'
* isInGroupDuringCompletePeriod
 - Returns: boolean
 - Function: Checks if student was in the given group in the specified period for the whole time.'
* numberOfDaysInGroupForThePeriod
 - Returns: int
 - Function: Count number of days student was in the given group in the specified period of time'
* getActiveGroupsAtAnyPointInThePeriod
 - Returns: List<String>
 - Function: Find the student's groups active at any point during specified period of time'
* getActiveGroupsAtDate
 - Returns: List<String>
 - Function: Find the student's active groups at the given date'
* getActiveGroupsDuringCompletePeriod
 - Returns: List<String>
 - Function: Find the student's groups active during whole specified period of time'
IMPORTANT REQUIREMENTS:
1. Each method usage must be explained in its full context
2. All data transformations must be detailed
3. Error handling for each method must be specified
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
```

```
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
 // Operation [number]: [name of calculation]
 // Formula: [complete mathematical expression]
 // Variables:
 // - [each variable and its purpose]
 // - [data types and units]
 // Steps:
 // 1. [detailed step-by-step breakdown]
 // 2. [intermediate calculations]
 // 3. [transformations applied]
 // Business Context: [why this calculation is needed]
 // Result: [meaning of the output]
 // Financial Impact: [effect on aid calculations]
 // Limitations: [any mathematical constraints]
 // If N/A, explicitly state: "ARITHMETIC: N/A"
ADDITIONAL REQUIREMENTS:
1. Complex Logic Sections:
 - Must break down multi-step operations
 - Explain interactions between components
```

```
   - Detail data flow and transformations
  2. Error Handling:
   - Document all error conditions
   - Explain recovery procedures
   - Detail logging and reporting
  3. API Interactions:
   - Document all platform method calls
   - Explain data transformations
   - Detail error handling
  4. Financial Aid Impact:
   - Explain effects on aid calculations
   - Document regulatory compliance
   - Detail student eligibility impact
  Example Section With All Aspects:
  // CONDITIONALS:
  // Check 1: Payment Period Status Validation
  // Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
  // Purpose: Ensure only valid, active payment periods are included
  // Logic:
  // - Retrieve all loan payment periods
  // - Filter based on status (!= "CANCELED")
  // - Compare start dates with reference date
  // Success Path: Period included in calculations
  // Failure Path: Period excluded from totals
  // Business Rules: Canceled periods must not affect grade level
  // Error Handling: Invalid periods logged and skipped
  // Financial Impact: Affects available loan amounts
  //
  // BOUNDARY CHECKS:
  // Check 1: Payment Period Date Validation
  // Source: Academic Calendar Configuration
  // Variables: period.getStartDate(), referenceDate
  // Validation:
  // - Compare period start date with reference date
  // - Ensure dates are not null
  // Edge Cases:
  // - Null dates are excluded
  // - Future dates are filtered
  // Error Handling: Invalid dates logged and skipped
  // Impact: Prevents future periods from affecting current calculation
  //
  // ARITHMETIC:
  // Operation 1: Payment Period Units Summation
  // Formula: sum(validPeriods.map(period => period.getUnits()))
  // Variables:
  // - period.getUnits(): Decimal, represents credit hours
  // - validPeriods: Array of filtered payment periods
  // Steps:
  // 1. Filter valid payment periods
  // 2. Extract units from each period
  // 3. Sum all units using reduce
  // Business Context: Calculate total completed units for grade level
  // Result: Total units affecting grade level
  // Financial Impact: Determines loan eligibility amounts
  // Limitations: Cannot exceed program maximum units
  ## Groovy Script for Analysis
  ```groovy
  //file:noinspection UnnecessaryQualifiedReference
  @groovy.transform.BaseScript(com.oracle.sfp.scripting.api.sdk.autocomplete.ExitCounselingWithdrawnCriteriaScript)
  package CHANGE_ME // this should be updated to your actual package name
  import com.oracle.sfp.scripting.api.*
  import com.oracle.sfp.scripting.api.util.*
  boolean returnValue = false;
  /* no API available for disbursements, and no way to separate out TEACH criteria based on current design */
  if (primaryProgram.getEnrollmentStatus() == "W")
  {
```
```

```
 returnValue = true;
}
return returnValue;
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.

# Professional Judgement Initiating Conditions

Name: professionalJudgementInitiatingConditions

Purpose: Determines if the student meets the criteria for professional judgement.

In Baseline Config: N

Return Type: Boolean

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "professionalJudgementInitiatingConditions"
Primary Function: Determines if a student meets the criteria to initiate a professional judgement process.
Expected Output Type: Boolean
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
allAwardYearReceivedDocuments
Purpose: Provides access to all documents received for the student.
Type: IMultiAwardYearReceivedDocsAPI
Available Methods:
* forAwardYear
 - Returns: IReceivedDocs
```

```
 - Function: Returns the collection of active received documents (IReceivedDocs) for an award year on an on-
demand,
cached basis.'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
```

```
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
isirRecord
Purpose: The ISIR associated to the professional judgement process.
Type: IIsirRecordAPI
Available Methods:
* getActiveIsirForAwardYear
 - Returns: Boolean
 - Function: Gets the active ISIR for award year status.'
* getAllFields
 - Returns: @NonNull Map<IsirField<?>,IIsirFieldValue<?>>
 - Function: Retrieves all field values (WARNING: can only be used with 2025+ ISIRs).'
* getAssumptions
 - Returns: List<IAssumptionAPI>
 - Function: Get the assumptions.'
* getAwardYear
 - Returns: String
 - Function: Gets the award year.'
* getCommentCodes
 - Returns: List<ICommentCodeAPI>
 - Function: Get the comment codes.'
* getCpsTransNumber
 - Returns: String
 - Function: Gets the CPS transaction number.'
* getCurrentSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's current SSN.'
* getDiscrepancies
 - Returns: List<IDiscrepancyAPI>
 - Function: Get the discrepancies.'
* getField
 - Returns: <T> @NonNull IIsirFieldValue<T>
 - Function: Retrieves the value of a given field (WARNING: can only be used with 2025+ ISIRs).'
* getFirstName
 - Returns: String
 - Function: Gets the student's first name.'
* getHighlightFlags
 - Returns: List<IHighlightFlagAPI>
 - Function: Get the highlight flags.'
* getIsirFieldValue
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValueOrNull
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValues
 - Returns: Map<String,String>
 - Function: Get a map containing all ISIR fields (WARNING: This method can only be used up to 2024 award
 year).'
* getIsirIdOriginalNameId
 - Returns: String
 - Function: Gets the student's original name ID.'
* getIsirIdSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's original SSN.'
* getIsirIdYearIndicator
 - Returns: String
 - Function: Gets the ISIR's year indicator.'
* getIsirStatus
 - Returns: String
 - Function: Gets the ISIR status.'
* getIsirStatusBeforeCorrection
 - Returns: String
 - Function: When ISIR Status is Corrected returns the previous ISIR Status.'
```

```
* getLastName
 - Returns: String
 - Function: Gets the student's last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the student's middle name.'
* getReceivedDate
 - Returns: Date
 - Function: Gets the system receipt date of the ISIR.'
* getRejectCodes
 - Returns: List<IRejectCodeAPI>
 - Function: Get the reject codes.'
* getSchoolCode1
 - Returns: String
 - Function: Gets the school code 1.'
* getSchoolCode10
 - Returns: String
 - Function: Gets the school code 10.'
* getSchoolCode2
 - Returns: String
 - Function: Gets the school code 2.'
* getSchoolCode3
 - Returns: String
 - Function: Gets the school code 3.'
* getSchoolCode4
 - Returns: String
 - Function: Gets the school code 4.'
* getSchoolCode5
 - Returns: String
 - Function: Gets the school code 5.'
* getSchoolCode6
 - Returns: String
 - Function: Gets the school code 6.'
* getSchoolCode7
 - Returns: String
 - Function: Gets the school code 7.'
* getSchoolCode8
 - Returns: String
 - Function: Gets the school code 8.'
* getSchoolCode9
 - Returns: String
 - Function: Gets the school code 9.'
* isCompatibleWithPackaging
 - Returns: boolean
 - Function: Checks if this ISIR satisfies minimal acceptance criteria for automatic Financial
Aid Package Calculation.'
* isUnverified
 - Returns: boolean
 - Function: Gets the ISIR verification status.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
```

```
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
receivedDocument
Purpose: Document causing the attempt to initiate the professional judgment process.
Type: IReceivedDocDocumentAPI
Available Methods:
* getComments
 - Returns: String
 - Function: No description available'
* getDocumentCode
 - Returns: String
 - Function: No description available'
* getDocumentDefn
 - Returns: IDocumentDefnAPI
 - Function: No description available'
* getDocumentFieldMetadataList
 - Returns: List<IDocumentFieldMetadataAPI>
 - Function: No description available'
* getDocumentFieldValue
 - Returns: String
 - Function: No description available'
* getDocumentOwner
 - Returns: String
 - Function: No description available'
* getDocumentOwnerTypeAsString
 - Returns: String
 - Function: No description available'
* getDocumentOwnerTypes
 - Returns: List<String>
 - Function: No description available'
* getDocumentSource
 - Returns: String
 - Function: No description available'
* getDocumentStatus
 - Returns: String
 - Function: No description available'
* getElectronicallyReceived
 - Returns: Boolean
 - Function: No description available'
* getExternalDocumentId
 - Returns: String
 - Function: No description available'
* getExternalStudentId
 - Returns: String
 - Function: No description available'
* getInZ
 - Returns: Date
 - Function: No description available'
* getOriginalDmsStatus
 - Returns: String
 - Function: No description available'
* getReceiptD
 - Returns: Date
 - Function: No description available'
* getResourceLink
 - Returns: String
 - Function: No description available'
* getReviewD
 - Returns: Date
 - Function: No description available'
* getReviewer
 - Returns: String
 - Function: No description available'
```

```
* getSubTypeCode
 - Returns: String
 - Function: No description available'
* getSubTypeIndicator
 - Returns: String
 - Function: No description available'
* isAcceptable
 - Returns: Boolean
 - Function: No description available'
* getAwardYear
 - Returns: String
 - Function: No description available'
receivedDocuments
Purpose: All student documents.
Type: IReceivedDocs
Available Methods:
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and Student, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code and owners, if it exists.'
* get
 - Returns: IReceivedDoc
 - Function: Return doc for specified code, owners, and subtype, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and Student, if it exists.'
* getDocumentsForDocCode
 - Returns: List<IReceivedDoc>
 - Function: Return list with doc or doc subtpyes for specified code and owners, if it exists.'
* getValidDocCodes
 - Returns: Set<String>
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: No description available'
* hasDoc
 - Returns: boolean
 - Function: Return true if doc exists with the right code and owners'
* noDocs
 - Returns: boolean
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
* put
 - Returns: void
 - Function: No description available'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
```

```
    - Returns: ISecureLocalDate
    - Function: Gets the date of birth.'
  * getDependencyStatus
    - Returns: String
    - Function: Gets the dependency status.'
  * getDeterminationDate
    - Returns: Date
    - Function: Gets the determination date.'
  * getDocuments
    - Returns: Map<String,IReceivedDoc>
    - Function: Gets the student's documents, indexed by document code.'
  * getEmail
    - Returns: String
    - Function: Gets the e-mail.'
  * getEnrollmentStatus
    - Returns: String
    - Function: Gets the enrollment status.'
  * getExternalStudentId
    - Returns: String
    - Function: Gets the external student ID (unique per student).'
  * getFinancialAidRequested
    - Returns: Boolean
    - Function: Gets the financial aid request status.'
  * getFirstName
    - Returns: String
    - Function: Gets the first name.'
  * getGender
    - Returns: String
    - Function: Gets the gender.'
  * getHighSchoolEquivalency
    - Returns: Boolean
    - Function: Gets the high school equivalency status.'
  * getLastName
    - Returns: String
    - Function: Gets the last name.'
  * getMiddleName
    - Returns: String
    - Function: Gets the middle name.'
  * getPersonId
    - Returns: Long
    - Function: Gets the person ID.'
  * getProgramType
    - Returns: String
    - Function: Gets the program type.'
  * getRecentFinancialAidHistoryRecords
    - Returns: IRecentFahRecordsAPI
    - Function: Gets the recent financial aid records.'
  * getSchoolCode
    - Returns: String
    - Function: Gets the school code.'
  * getSchoolEmployee
    - Returns: Boolean
    - Function: Gets the school employee status.'
  * getSchoolEmployeeDependent
    - Returns: Boolean
    - Function: Gets the school employee dependent status.'
  * getSsn
    - Returns: ISocialSecurityNumber
    - Function: Gets the SSN.'
  * getStudentId
    - Returns: Long
    - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
  IMPORTANT REQUIREMENTS:
  1. Each method usage must be explained in its full context
  2. All data transformations must be detailed
  3. Error handling for each method must be specified
```

```
4. Business implications of method results must be clear
## COMMENTING REQUIREMENTS
Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
FILE-LEVEL DOCUMENTATION REQUIREMENTS:
1. File Header (Using /** ... */ format):
 MUST INCLUDE (15+ lines):
 - Script name and version
 - Primary purpose and functionality
 - All key calculations performed
 - Critical business rules enforced
 - Boundary conditions and limits
 - Financial aid implications
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
 // Operation [number]: [name of calculation]
 // Formula: [complete mathematical expression]
 // Variables:
 // - [each variable and its purpose]
 // - [data types and units]
 // Steps:
 // 1. [detailed step-by-step breakdown]
 // 2. [intermediate calculations]
 // 3. [transformations applied]
 // Business Context: [why this calculation is needed]
 // Result: [meaning of the output]
 // Financial Impact: [effect on aid calculations]
 // Limitations: [any mathematical constraints]
 // If N/A, explicitly state: "ARITHMETIC: N/A"
```

```
ADDITIONAL REQUIREMENTS:
1. Complex Logic Sections:
 - Must break down multi-step operations
 - Explain interactions between components
 - Detail data flow and transformations
2. Error Handling:
 - Document all error conditions
 - Explain recovery procedures
 - Detail logging and reporting
3. API Interactions:
 - Document all platform method calls
 - Explain data transformations
 - Detail error handling
4. Financial Aid Impact:
 - Explain effects on aid calculations
 - Document regulatory compliance
 - Detail student eligibility impact
Example Section With All Aspects:
// CONDITIONALS:
// Check 1: Payment Period Status Validation
// Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
N/A - There is no baseline config. This is a placeholder.
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
```

```
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# Professional Judgement Required Documents

Name: professionalJudgementRequiredDocuments

Purpose: Determines the documents required for professional judgement.

In Baseline Config: No

Return Type: Void

```
# Oracle SFP Platform Script Documentation Generator
Your task is to create comprehensive, contextual comments for this Oracle SFP Platform Groovy script.
CRITICAL REQUIREMENTS:
1. Follow exact comment format specified
2. Include all three aspects for every section
3. Mark non-applicable aspects as "N/A"
4. Provide business context and financial impact
5. Detail all API interactions and data flows
## Script Purpose and Context
The script being analyzed is: "professionalJudgementRequiredDocuments"
Primary Function: Determines the documents that must be requested for the professional judgement process.
Expected Output Type: Void
This script is part of the Oracle SFP Platform ecosystem and interacts with platform-specific APIs and data
 structures.
You must analyze this script in the context of financial aid processing workflows, ensuring compliance with:
1. Federal financial aid regulations
2. Academic progress requirements
3. Institutional policies
4. Program-specific rules
Your analysis must detail:
1. All conditional logic and its business implications
2. All boundary checks and their validation purposes
3. All calculations and their financial aid impact
4. Error handling and data validation
5. Integration points with platform APIs
## Available Platform Methods and Integrations
The script has access to the following Oracle SFP Platform integrations.
When commenting the code, you MUST explain how these specific methods and bindings are being used:
currentIsir
Purpose: The ISIR associated to the professional judgement process.
Type: IIsirRecordAPI
Available Methods:
* getActiveIsirForAwardYear
 - Returns: Boolean
 - Function: Gets the active ISIR for award year status.'
* getAllFields
 - Returns: @NonNull Map<IsirField<?>,IIsirFieldValue<?>>
 - Function: Retrieves all field values (WARNING: can only be used with 2025+ ISIRs).'
* getAssumptions
 - Returns: List<IAssumptionAPI>
 - Function: Get the assumptions.'
```

```
* getAwardYear
 - Returns: String
 - Function: Gets the award year.'
* getCommentCodes
 - Returns: List<ICommentCodeAPI>
 - Function: Get the comment codes.'
* getCpsTransNumber
 - Returns: String
 - Function: Gets the CPS transaction number.'
* getCurrentSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's current SSN.'
* getDiscrepancies
 - Returns: List<IDiscrepancyAPI>
 - Function: Get the discrepancies.'
* getField
 - Returns: <T> @NonNull IIsirFieldValue<T>
 - Function: Retrieves the value of a given field (WARNING: can only be used with 2025+ ISIRs).'
* getFirstName
 - Returns: String
 - Function: Gets the student's first name.'
* getHighlightFlags
 - Returns: List<IHighlightFlagAPI>
 - Function: Get the highlight flags.'
* getIsirFieldValue
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValueOrNull
 - Returns: String
 - Function: Gets the value of an ISIR field (WARNING: This method can only be used up to 2024 award year).'
* getIsirFieldValues
 - Returns: Map<String,String>
 - Function: Get a map containing all ISIR fields (WARNING: This method can only be used up to 2024 award
 year).'
* getIsirIdOriginalNameId
 - Returns: String
 - Function: Gets the student's original name ID.'
* getIsirIdSsn
 - Returns: ISocialSecurityNumber
 - Function: Gets the student's original SSN.'
* getIsirIdYearIndicator
 - Returns: String
 - Function: Gets the ISIR's year indicator.'
* getIsirStatus
 - Returns: String
 - Function: Gets the ISIR status.'
* getIsirStatusBeforeCorrection
 - Returns: String
 - Function: When ISIR Status is Corrected returns the previous ISIR Status.'
* getLastName
 - Returns: String
 - Function: Gets the student's last name.'
* getMiddleName
 - Returns: String
 - Function: Gets the student's middle name.'
* getReceivedDate
 - Returns: Date
 - Function: Gets the system receipt date of the ISIR.'
* getRejectCodes
 - Returns: List<IRejectCodeAPI>
 - Function: Get the reject codes.'
* getSchoolCode1
 - Returns: String
 - Function: Gets the school code 1.'
* getSchoolCode10
 - Returns: String
```

```
 - Function: Gets the school code 10.'
* getSchoolCode2
 - Returns: String
 - Function: Gets the school code 2.'
* getSchoolCode3
 - Returns: String
 - Function: Gets the school code 3.'
* getSchoolCode4
 - Returns: String
 - Function: Gets the school code 4.'
* getSchoolCode5
 - Returns: String
 - Function: Gets the school code 5.'
* getSchoolCode6
 - Returns: String
 - Function: Gets the school code 6.'
* getSchoolCode7
 - Returns: String
 - Function: Gets the school code 7.'
* getSchoolCode8
 - Returns: String
 - Function: Gets the school code 8.'
* getSchoolCode9
 - Returns: String
 - Function: Gets the school code 9.'
* isCompatibleWithPackaging
 - Returns: boolean
 - Function: Checks if this ISIR satisfies minimal acceptance criteria for automatic Financial
Aid Package Calculation.'
* isUnverified
 - Returns: boolean
 - Function: Gets the ISIR verification status.'
documentRequest
Purpose: Binding that shall be populated by the script with the requested documents.
Type: IIsirRequiredDocsAPI
Available Methods:
* addAnyDocuments
 - Returns: void
 - Function: Adds a list of documents, any one in the list is required.'
* addAnyDocuments
 - Returns: void
 - Function: Adds a list of documents, any one in the list is required.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* addDocument
 - Returns: void
 - Function: Adds a required document.'
* any
 - Returns: IIsirRequiredDocsAPI
 - Function: No description available'
* clear
 - Returns: void
 - Function: Clears all docs.'
* doc
 - Returns: IIsirRequiredDocsAPI
 - Function: No description available'
* getAdditionalDocRequestMessages
 - Returns: List<String>
 - Function: No description available'
* getValidDocCodes
 - Returns: Set<String>
 - Function: No description available'
```

```
* hasDocument
 - Returns: boolean
 - Function: Returns true if the document has been added, either as a regular request
or an 'any' request.'
* hasDocument
 - Returns: boolean
 - Function: Returns true if the document has been added, either as a regular request
or an 'any' request.'
* isEmpty
 - Returns: boolean
 - Function: Checks if these requirements are empty.'
* setValidDocCodes
 - Returns: void
 - Function: No description available'
enrollment
Purpose: Information for the enrollment this script is being evaluated for.
Type: IEnrollmentAPI
Available Methods:
* belongsTo
 - Returns: boolean
 - Function: Checks if this enrollment belongs to a student.'
* getBindings
 - Returns: IEnrollmentBindingsAPI
 - Function: Gets bindings for this enrollment.'
* getEnrollmentStatus
 - Returns: String
 - Function: Gets the enrollment status.'
* getProgramStartDate
 - Returns: LocalDate
 - Function: Gets the program start date.'
* isEffective
 - Returns: boolean
 - Function: Checks if this enrollment is effective.'
enrollments
Purpose: Enrollments information.
Type: IEnrollmentsAPI
Available Methods:
* getHigherPriorityThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of enrollments with higher priority than a given enrollment.'
* getOtherThan
 - Returns: List<IEnrollmentAPI>
 - Function: Gets a list of the enrollments that are different from this enrollment, no
matter the priority.'
helper
Purpose: Helper interface with utility methods.
Type: IGroovyScriptHelper
Available Methods:
* filterCourses
 - Returns: ICourseFilterAPI
 - Function: No description available'
* getAwardYear
 - Returns: IAwardYearHelperAPI
 - Function: Returns Award Year-specific helper.'
* getIntegerFromSignedIsirNumber
 - Returns: Integer
 - Function: Parses a number represented in ISIR format to Integer.'
* getIsirDate
 - Returns: Date
 - Function: No description available'
* getIsirDateFormatStr
 - Returns: String
 - Function: No description available'
* getSignedIsirNumberFromInteger
 - Returns: String
 - Function: No description available'
```

```
* isirRecordsDifferForListedFields
 - Returns: boolean
 - Function: No description available'
* mapRanges
 - Returns: IRangeMapStarterAPI
 - Function: No description available'
* newAwardYearPeriod
 - Returns: IAwardYearAPI
 - Function: Returns the Award Year having a given end year.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns the IPeriod for one single day.'
* newPeriod
 - Returns: IPeriod
 - Function: Returns a IPeriod given its start and end dates.'
log
Purpose: Provides logging capability to scripts.
Type: IScriptLoggerAPI
Available Methods:
* debug
 - Returns: boolean
 - Function: No description available'
* debug
 - Returns: void
 - Function: No description available'
* error
 - Returns: void
 - Function: No description available'
* info
 - Returns: boolean
 - Function: No description available'
* info
 - Returns: void
 - Function: No description available'
* warn
 - Returns: void
 - Function: No description available'
student
Purpose: Student information.
Type: IFasStudentAPI
Available Methods:
* getAddressState
 - Returns: String
 - Function: Gets the federal state in the student's main mailing address, as a two-letter acronym.'
* getBahIndicator
 - Returns: Boolean
 - Function: Gets the school employee dependet status.'
* getCampusCode
 - Returns: String
 - Function: Gets the campus code.'
* getCompletionDate
 - Returns: Date
 - Function: Gets the completion date.'
* getDateOfBirth
 - Returns: ISecureLocalDate
 - Function: Gets the date of birth.'
* getDependencyStatus
 - Returns: String
 - Function: Gets the dependency status.'
* getDeterminationDate
 - Returns: Date
 - Function: Gets the determination date.'
* getDocuments
 - Returns: Map<String,IReceivedDoc>
 - Function: Gets the student's documents, indexed by document code.'
* getEmail
```

```
    - Returns: String
    - Function: Gets the e-mail.'
  * getEnrollmentStatus
    - Returns: String
    - Function: Gets the enrollment status.'
  * getExternalStudentId
    - Returns: String
    - Function: Gets the external student ID (unique per student).'
  * getFinancialAidRequested
    - Returns: Boolean
    - Function: Gets the financial aid request status.'
  * getFirstName
    - Returns: String
    - Function: Gets the first name.'
  * getGender
    - Returns: String
    - Function: Gets the gender.'
  * getHighSchoolEquivalency
    - Returns: Boolean
    - Function: Gets the high school equivalency status.'
  * getLastName
    - Returns: String
    - Function: Gets the last name.'
  * getMiddleName
    - Returns: String
    - Function: Gets the middle name.'
  * getPersonId
    - Returns: Long
    - Function: Gets the person ID.'
  * getProgramType
    - Returns: String
    - Function: Gets the program type.'
  * getRecentFinancialAidHistoryRecords
    - Returns: IRecentFahRecordsAPI
    - Function: Gets the recent financial aid records.'
  * getSchoolCode
    - Returns: String
    - Function: Gets the school code.'
  * getSchoolEmployee
    - Returns: Boolean
    - Function: Gets the school employee status.'
  * getSchoolEmployeeDependent
    - Returns: Boolean
    - Function: Gets the school employee dependent status.'
  * getSsn
    - Returns: ISocialSecurityNumber
    - Function: Gets the SSN.'
  * getStudentId
    - Returns: Long
    - Function: Gets the student ID (or enrollment ID, unique per enrollment).'
  IMPORTANT REQUIREMENTS:
  1. Each method usage must be explained in its full context
  2. All data transformations must be detailed
  3. Error handling for each method must be specified
  4. Business implications of method results must be clear
  ## COMMENTING REQUIREMENTS
  Your task is to create COMPREHENSIVE, CONTEXTUAL comments for this Oracle SFP Platform script.
  FILE-LEVEL DOCUMENTATION REQUIREMENTS:
  1. File Header (Using /** ... */ format):
    MUST INCLUDE (15+ lines):
    - Script name and version
    - Primary purpose and functionality
    - All key calculations performed
    - Critical business rules enforced
    - Boundary conditions and limits
    - Financial aid implications
```

```
 - Required inputs and expected outputs
 - Error handling approach
 - Integration points
 - Compliance considerations
SECTION-LEVEL DOCUMENTATION REQUIREMENTS:
Every logical code section must include detailed comments for ALL three aspects:
1. CONDITIONALS Section:
 Format:
 // CONDITIONALS:
 // Check [number]: [specific condition being evaluated]
 // Variables: [list of variables and methods involved]
 // Purpose: [detailed business reason for this check]
 // Logic:
 // - [step-by-step breakdown of the condition]
 // - [how variables are used]
 // - [any transformations applied]
 // Success Path: [what happens when condition is true]
 // Failure Path: [what happens when condition is false]
 // Business Rules: [specific rules being enforced]
 // Error Handling: [how errors are managed]
 // Financial Impact: [effect on aid calculations]
 // If N/A, explicitly state: "CONDITIONALS: N/A"
2. BOUNDARY CHECKS Section:
 Format:
 // BOUNDARY CHECKS:
 // Check [number]: [specific limit being enforced]
 // Source: [where boundary values come from]
 // Variables: [variables and methods involved]
 // Validation:
 // - [how boundaries are checked]
 // - [validation method details]
 // Edge Cases:
 // - [specific edge cases handled]
 // - [how each is managed]
 // Error Handling: [boundary violation management]
 // Impact: [effect of boundary violations]
 // If N/A, explicitly state: "BOUNDARY CHECKS: N/A"
3. ARITHMETIC Section:
 Format:
 // ARITHMETIC:
 // Operation [number]: [name of calculation]
 // Formula: [complete mathematical expression]
 // Variables:
 // - [each variable and its purpose]
 // - [data types and units]
 // Steps:
 // 1. [detailed step-by-step breakdown]
 // 2. [intermediate calculations]
 // 3. [transformations applied]
 // Business Context: [why this calculation is needed]
 // Result: [meaning of the output]
 // Financial Impact: [effect on aid calculations]
 // Limitations: [any mathematical constraints]
 // If N/A, explicitly state: "ARITHMETIC: N/A"
ADDITIONAL REQUIREMENTS:
1. Complex Logic Sections:
 - Must break down multi-step operations
 - Explain interactions between components
 - Detail data flow and transformations
2. Error Handling:
 - Document all error conditions
 - Explain recovery procedures
 - Detail logging and reporting
3. API Interactions:
 - Document all platform method calls
 - Explain data transformations
```

```
 - Detail error handling
4. Financial Aid Impact:
 - Explain effects on aid calculations
 - Document regulatory compliance
 - Detail student eligibility impact
Example Section With All Aspects:
// CONDITIONALS:
// Check 1: Payment Period Status Validation
// Variables: getLoanPaymentPeriods(), period.getStatus(), period.getStartDate()
// Purpose: Ensure only valid, active payment periods are included
// Logic:
// - Retrieve all loan payment periods
// - Filter based on status (!= "CANCELED")
// - Compare start dates with reference date
// Success Path: Period included in calculations
// Failure Path: Period excluded from totals
// Business Rules: Canceled periods must not affect grade level
// Error Handling: Invalid periods logged and skipped
// Financial Impact: Affects available loan amounts
//
// BOUNDARY CHECKS:
// Check 1: Payment Period Date Validation
// Source: Academic Calendar Configuration
// Variables: period.getStartDate(), referenceDate
// Validation:
// - Compare period start date with reference date
// - Ensure dates are not null
// Edge Cases:
// - Null dates are excluded
// - Future dates are filtered
// Error Handling: Invalid dates logged and skipped
// Impact: Prevents future periods from affecting current calculation
//
// ARITHMETIC:
// Operation 1: Payment Period Units Summation
// Formula: sum(validPeriods.map(period => period.getUnits()))
// Variables:
// - period.getUnits(): Decimal, represents credit hours
// - validPeriods: Array of filtered payment periods
// Steps:
// 1. Filter valid payment periods
// 2. Extract units from each period
// 3. Sum all units using reduce
// Business Context: Calculate total completed units for grade level
// Result: Total units affecting grade level
// Financial Impact: Determines loan eligibility amounts
// Limitations: Cannot exceed program maximum units
## Groovy Script for Analysis
```groovy
N/A - There is no baseline config. This is a placeholder.
```
## COMMENT GENERATION REQUIREMENTS:
1. Add comprehensive comments following the exact format specified
2. Every logical section must include all three aspects:
 - CONDITIONALS
 - BOUNDARY CHECKS
 - ARITHMETIC
3. Mark any non-applicable aspect as "N/A" explicitly
4. Ensure line-by-line explanation of complex logic
5. Detail all financial aid implications
DO NOT modify the code itself. Add comments only.
REMEMBER:
- Comments must be exhaustively detailed
- Every logical section needs all three aspects
- Financial aid impact must be clear
- Business rules must be explicit
```

```
Return ONLY the fully commented Groovy script. No other text or explanations.
```

# 4 Troubleshoot Common Issues

You might come across some issues when using these prompts. Here are some common issues and how to resolve them.

## Incomplete Documentation

If you receive incomplete documentation from the generative AI tool, check that you copied all the sections of the prompt. Check for formatting issues as well.

## Incorrect Context

If the resulting context from the generative AI tool is incorrect, make sure you're using the correct prompt for your script. Also check that the script type matches the configuration.

## Missing Method Information

If methods are missing from the generative AI tool output, verify with Java Docs and baseline configuration. You can also add any custom methods to the prompt.