Oracle® Student System

Open Interfaces User's Guide

Release 11i

Part No. B10522-02

April 2003

ORACLE®

Oracle Student System Open Interfaces User’s Guide, Release 11*i*

Part No. B10522-02

Primary Author: Sankara Rajanala

Contributor: Shailaja Babanagar, Satish Bhat, Judy Boles, Ray Bradbury, Mary Brilliant, Scott Brown, Venkata Chamakura, Brian Chan, Maxine Chan, Sita Chodavarapu, Amitava Sen Choudhuri, Kit Craig, Phil Cram, Balachander Devakumar, Joydeep Dey, Scott Eldredge, Srini Gaddamadugu, Amit Gairola, Boe Heynike, Bill Hollowsky, Andrea Hu, Geetha Iyer, Nagaraju Jammidiguntla, Alison Jenkins, Bharathy Kaliappan, Syam Krishnan, Gustavus Kundahl, Lee Lay, Joe LeCluyse, Benjamin Lee, Sarah Macdonald, Prasanti Madireddy, Tultul Majumder, Karthikeyan Mohan, Rinku Mohapatra, Annamalai Muthu, Margaret Neary, Srinivasasarma Nemani, Elizabeth Newell, Vinod Palcharla, Nancy Raabe, Barbara Randlett, Ramesh Rengarajan, Simran Sawhney, Neelam Singh, Dan Stockwell, Carlton R. Stroud, Carl Vadnais, Venkata Voruganti, Sasi Yerrabolu

Contents

1 Introduction

Part I Legacy Data Import

2 Legacy Data Import Processes

Part II Admissions Module

3 Admissions Import Process

4 Institutions Import Process

5 Probability Values Import Process

6 Academic Index Import Process

7 Test Score Conversion and Import Process

8 Automatic Batch Admission Decisions Import Process

9 Offer Response Import Process

Part III Enrollments Module

10 Enrollment Registration Update Process

11 Interactive Voice Response Process

Part IV Financial Aid Module

12 Student Employment Payroll Data Upload Process

Part V Student Finance Module

13 External Charges Import Process

14 Import and Calculate Ancillary Charges Process

15 Credits API

16 Student Refund Transfer Process

Part VII Academic Records Module

20 Degree Audit Data Exchange Process

Part VIII HESA Data Import

21 HESA Data Import Process

Part IX References

A Admission Import Process Error Codes

B Admission Import Source Category Tables

Index

Send Us Your Comments

Oracle Student System Open Interfaces User's Guide, Release 11*i*

Part No. B10522-02

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the chapter, section, and page number (if available). You can send comments to:

Oracle Corporation
Oracle Student System
500 Oracle Parkway
Redwood City, CA 94065
U.S.A.

If you would like a reply, please give your name, address, and telephone number below.

If you have problems with the software, please contact Oracle Support Services.

Preface

The *Oracle Student System Open Interfaces User's Guide* provides information on how to use Oracle Student System interfaces.

The following sections are included in this preface:

- Audience for this Guide
- Conventions
- Documentation Accessibility
- Other Information Sources
- Navigation Paths
- Training and Support
- Do Not Use Database Tools to Modify Oracle Applications Data
- About Oracle
- Documentation Sales
- Feedback

Audience for this Guide

Welcome to Release 11*i* of the *Oracle Student System Open Interfaces User's Guide*.

This guide assumes users have a working knowledge of the following:

- Principles and customary practices of the business area
- Oracle Student System

Oracle suggests that users who have never used Oracle Student System attend one or more of the Oracle Student System training classes available through Oracle University.

- Oracle Applications graphical user interface

  To learn more about the Oracle Applications graphical user interface, read the *Oracle Applications User's Guide*.

See Other Information Sources for more information about Oracle Applications product information.

Conventions

The following conventions are observed:

- Conventions
- Usage Conventions
- References

Special Conventions

The following special conventions are observed:

Table 1 Special Conventions

bold	Bold type denotes buttons, as in the following example: Click **Unit Details** to transfer units.
Titlecase	Titlecase text denotes Oracle keywords and predefined system statuses, as in the following example: Confirmed
Note:	Notes alert users to the following type of information in this guide: **Note:** Notes alert users to key points to consider when using a feature.
WARNING:	Warnings alert users to the following type of information in this guide: **WARNING:** Warnings highlight text that warns of actions that could result in loss of data or incorrect processing.

Usage Conventions

The following usage conventions are observed:

Table 2 Usage Conventions

Attach file	Indicates that users should click the Attachment icon to attach a document to a record. For information on attachments, see About Attachments and Attachments Window, *Oracle Applications User's Guide*.
Save your work	Indicates users should save your work and close the window. To close the window users can either use the **File - Close Form** command or click on the x in the upper right-hand corner. **Note**: The **File - Close Form** command produces different results depending on the product and platform in use. For example, sometimes it closes only one window; at other times, it closes all open windows. Users must familiarize themselves with how the command behaves in their own environments.
Descriptions of Graphics	Textual descriptions accompany all graphics that appear in this guide.
Query appropriate data	Indicates users should query data in the window using **View - Find** or **View - Query By Example - Enter**. For information on entering and querying data, see Entering and Querying Data, *Oracle Applications User's Guide.*

References

All references to specific chapters refer to chapters in this guide unless otherwise noted.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle Corporation is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/

Accessibility of Code Examples in Documentation

JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

Other Information Sources

Users can choose from many sources of information, including online documentation, training, and support services, to increase their knowledge and understanding of Oracle Student System.

When this guide refers to other Oracle Applications documentation, use only the Release 11*i* versions of those guides.

Related User’s Guides

Oracle Student System shares business and setup information with other Oracle Applications products. Users may want to refer to other user’s guides when setting up and using Oracle Student System.

Read the guides online by choosing Library from the expandable menu on the HTML help window, by reading from the Oracle Applications Document Library CD included in the media pack, or using a Web browser with a URL provided by the system administrator.

Purchase printed guides, if required, from the Oracle Store at http://oraclestore.oracle.com.

Guides Related to All Products

Oracle Applications User’s Guide

This guide explains how to enter data, query, run reports, and navigate using the graphical user interface available with this release of Oracle Student System and any other Oracle Applications products. This guide also includes information on setting user profiles, as well as running and reviewing reports and concurrent processes.

Access this user’s guide by choosing Getting Started with Oracle Applications from any Oracle Applications help file.

User’s Guides Related to This Product

This guide contains references to the following Oracle publications:

- *Configuring, Reporting, and System Administration in HRMS*
- *Managing People Using Oracle HRMS*
- *Multiple Organizations in Oracle Applications*
- *Oracle Applications Desktop Integrator User’s Guide*
- *Oracle Applications Flexfields Guide*
- *Oracle Applications System Administrator’s Guide*
- *Oracle Bill of Materials User’s Guide*
- *Oracle CRM Application Foundation Implementation Guide*
- *Oracle CRM Application Foundation Concepts and Procedures*
- *Oracle General Ledger User’s Guide*
- *Oracle Inventory User’s Guide*
- *Oracle Marketing Online Implementation Guide*
- *Oracle Marketing Online Concepts and Procedures*
- *Oracle Payables User’s Guide*
- *Oracle Receivables User’s Guide*
- *Oracle Student System Implementation Guide*
- *Oracle Student System Open Interfaces User’s Guide*
- *Oracle Student System User’s Guide*
- *Oracle TeleSales Implementation Guide*
- *Oracle TeleSales Concepts and Procedures*
- *Oracle Workflow Guide*
- *Using Oracle HRMS - The Fundamentals*

Installation and System Administration

Oracle Applications Concepts

This guide provides an introduction to the concepts, features, technology stack, architecture, and terminology for Oracle Applications, Release 11*i*. It is a useful first book to read before installing Oracle Applications. This guide also introduces the concepts behind Applications-wide features such as Business Intelligence (BIS), languages and character sets, and Self Service Web Applications.

Installing Oracle Applications

This guide provides instructions for managing the installation of Oracle Applications products. In Release 11*i*, much of the installation process is handled using Oracle Rapid Install, which minimizes the time to install Oracle Applications, the Oracle8 technology stack, and the Oracle8*i* Server technology stack by automating many of the required steps. This guide contains instructions for using Oracle Rapid Install and lists the tasks needed to complete an installation. Use this guide in conjunction with individual product user's guides and implementation guides.

Upgrading Oracle Applications

Refer to this guide when upgrading Oracle Applications Release 10.7 or Release 11.0 products to Release 11*i*. This guide describes the upgrade process and lists database and product-specific upgrade tasks. To upgrade to Release 11*i*, users must be at Release 10.7, in NCA, SmartClient, or character mode, or Release 11.0. Users cannot upgrade to Release 11*i* directly from releases prior to 10.7.

Maintaining Oracle Applications

Use this guide to run various AD utilities, such as AutoUpgrade, AutoPatch, AD Administration, AD Controller, AD Relink, License Manager, and others. The guide contains how-to steps, screenshots, and other information needed to run the AD utilities. This guide also provides information on maintaining the Oracle Applications file system and database.

Oracle Applications System Administrator's Guide

The guide provides planning and reference information for the Oracle Applications system administrator. The guide contains information on how to define security, customize menus and online help, and manage concurrent processing.

Oracle Alert User’s Guide

This guide explains how to define periodic and event alerts to monitor the status of Oracle Applications data.

Oracle Applications Developer’s Guide

This guide contains the coding standards followed by the Oracle Applications development staff. It describes the Oracle Application Object Library components needed to implement the Oracle Applications user interface (UI) described in the *Oracle Applications User Interface Standards for Forms-Based Products*. It also provides information to help users build custom Oracle Forms Developer 6*i* forms so that they integrate with Oracle Applications.

Oracle Applications User Interface Standards for Forms-Based Products

This guide contains the UI standards followed by the Oracle Applications development staff. It describes the UI for Oracle Applications products and how to apply this UI to the design of an application built by using Oracle Forms.

Implementation Documentation

Oracle Student System Implementation Guide

The Oracle Student System Implementation Guide describes setup for integrations with other Oracle products and features in addition to providing setup information for Oracle Student System itself. You use this guide for information about setting up your codes, types, profile options, and institution information in Oracle Student System.

Oracle Applications Product Update Notes

Use this guide as a reference for upgrading an installation of Oracle Applications. It provides a history of the changes to individual Oracle Applications products between Release 11.0 and Release 11*i*. It includes new features, enhancements, and changes made to database objects, profile options, and seed data for this interval.

Oracle Workflow Administrator's Guide

This guide explains how to complete the setup steps necessary for any Oracle Applications product that includes workflow-enabled processes, as well as how to monitor the progress of runtime workflow processes.

Oracle Workflow Developer's Guide

This guide explains how to define new workflow business processes and customize existing Oracle Applications-embedded workflow processes. It also describes how to define and customize business events and event subscriptions.

Oracle Workflow User's Guide

This guide describes how Oracle Applications users can view and respond to workflow notifications and monitor the progress of their workflow processes.

Oracle Workflow API Reference

This guide describes the APIs provided for developers and administrators to access Oracle Workflow.

Oracle Applications Flexfields Guide

This guide provides flexfields planning, setup, and reference information for the Oracle Student System implementation team, as well as for users responsible for the ongoing maintenance of Oracle Applications product data. This manual also provides information on creating custom reports on flexfields data.

Oracle eTechnical Reference Manuals

Each eTechnical Reference Manual (eTRM) contains database diagrams and a detailed description of database tables, forms, reports, and programs for a specific Oracle Applications product. This information helps users convert data from existing applications, integrate Oracle Applications data with non-Oracle applications, and write custom reports for Oracle Applications products. Oracle eTRM is available on MetaLink.

Oracle Manufacturing APIs and Open Interfaces Manual

This manual contains up-to-date information about integrating with other Oracle Manufacturing applications and with other systems. The guide includes APIs and open interfaces found in Oracle Manufacturing.

Oracle Order Management Suite APIs and Open Interfaces Manual

The manual contains up-to-date information about integrating with other Oracle Manufacturing applications and with other systems. The guide includes APIs and open interfaces found in Oracle Order Management Suite.

Oracle Applications Message Reference Manual

This manual describes all Oracle Applications messages. The guide is available in HTML format on the documentation CD-ROM for Release 11*i*.

Navigation Paths

Navigation paths for windows in Oracle Student System are documented for the system as they are shipped. If responsibilities are changed after installation the documented navigation paths may not be correct.

Training and Support

Training

Users have a choice of educational environments. They can attend courses offered by Oracle University at any one of our many Education Centers, or can arrange for our trainers to teach at their facility, or can employ the Oracle Learning Network (OLN), Oracle University's online education utility. In addition, Oracle training professionals can tailor standard courses or develop custom courses to meet users' needs. For example, users can employ their own organizational structure, terminology, and data as examples in a customized training session delivered at their own facility.

Support

From on-site support to central support, our team of experienced professionals provides the help and information needed to keep Oracle Student System working for all users. This team includes the technical representative, account manager, and Oracle's large staff of consultants and support specialists with expertise in users' specific business areas, managing an Oracle server, and users' hardware and software environments.

Do Not Use Database Tools to Modify Oracle Applications Data

We STRONGLY RECOMMEND that users never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle Applications tables, unless otherwise instructed.

Oracle Corporation provides powerful tools users can employ to create, store, change, retrieve, and maintain information in an Oracle database. But if users employ tools such as SQL*Plus to modify Oracle Applications data, they risk destroying the integrity of the data and lose the ability to audit changes to the data.

Because Oracle Applications tables are interrelated, any change made using Oracle Applications can update many tables at once. But when users modify Oracle Applications data using anything other than Oracle Applications, users might change a row in one table without making corresponding changes in related tables. If the tables get out of synchronization with each other, users risk retrieving erroneous information and unpredictable results throughout Oracle Applications.

When users employ Oracle Applications to modify the data, Oracle Applications automatically checks that the changes are valid. Oracle Applications also keeps track of who changes the information. But if users enter information into database tables using database tools, users can store invalid information. Users also lose the ability to track who has changed the information because SQL*Plus and other database tools do not keep a record of changes.

About Oracle

Oracle Corporation develops and markets an integrated line of software products for database management, applications development, decision support, and office automation, as well as Oracle Applications, an integrated suite of more than 160 software modules for financial management, supply chain management, manufacturing, project systems, human resources, and customer relationship management.

Oracle products are available for mainframes, minicomputers, personal computers, network computers, and personal digital assistants, enabling organizations to integrate different computers, different operating systems, different networks, and even different database management systems, into a single, unified computing and information resource.

Oracle is the world’s leading supplier of software for information management, and the world’s second largest software company. Oracle offers its database, tools, and application products, along with related consulting, education, and support services, in over 145 countries around the world.

Documentation Sales

To order hard copy documentation, go to the Oracle Store at http://oraclestore.oracle.com.

Support sales contact information by region and country is available at http://www.oracle.com/support/index.html?contact.html.

Feedback

Thank you for using Oracle Student System and this user’s guide.

Oracle values comments and feedback. Users can employ the Reader’s Comment Form provided to explain what they like or dislike about Oracle Student System or this user’s guide. Mail comments to the following address:

Oracle Student System Documentation Manager

Oracle Corporation

500 Oracle Parkway

Redwood Shores, CA 94065

U.S.A.

1

Introduction

This chapter describes how the *Oracle Student System Open Interfaces User's Guide* is organized. The following sections are in this chapter:

- Overview
- Legacy Data Import
- Admissions Module
- Enrollments Module
- Financial Aid Module
- Student Finance Module
- Program Structure and Planning Module
- Academic Records Module
- HESA Data Import
- References

Overview

The *Oracle Student System Open Interfaces User's Guide* contains information needed to understand and use Oracle Student System open interface tables and an API.

This guide is divided into the following parts:

- Legacy Data Import
- Admissions Module
- Enrollments Module

- Financial Aid Module
- Student Finance Module
- Program Structure and Planning Module
- Academic Records Module
- HESA Data Import
- References

Legacy Data Import

The legacy data import process is used to load legacy data to the Program Structure and Planning, Enrollments, and Records subsystems or Oracle Student System. The following chapter describes legacy data import process and the validations performed during the process:

- Legacy Data Import Process

Admissions Module

The following chapters describe how to import person and admission information into the Admissions module of Oracle Student System:

- Admissions Import Process
- Institutions Import Process

The following chapters describe how to process and transfer applicant information within the Admissions module of Oracle Student System:

- Probability Values Import Process
- Academic Index Import Process
- Test Score Conversion and Import Process
- Automatic Batch Admission Decisions Import Process

Enrollments Module

The following chapters describe how to import enrollment registration data and interface with interactive voice response systems used for enrollments:

- Enrollment Registration Update Process
- Interactive Voice Response Process

Financial Aid Module

The following chapter describes how to upload work-study payroll data from an internal or external Human Resources or Payroll system to the target table in the Financial Aid module of Oracle Student System:

- Employment Payroll Data Upload Process

Student Finance Module

The following chapters describe how to transfer student financial data within the Student Finance module of Oracle Student System:

- External Charges Import Process
- Import and Calculate Ancillary Charges Process
- Credits API
- Student Refund Transfer Process
- Posting Interface Process
- Sponsorship Assignment API

Program Structure and Planning Module

The following chapter describes how to transfer Unit Section Occurrence information from Oracle Student System to a third party scheduling application and back to Oracle Student System:

- Scheduling Interface Process

Academic Records Module

The following chapter describes how to transfer degree audit data from Oracle Student System to a third party degree audit application and back to Oracle Student System.

- Degree Audit Data Exchange Process

HESA Data Import

The following chapter describes how to import HESA data into Oracle Student System:

- HESA Data Import Process

This process is meant for users of Oracle Student System for the United Kingdom.

Legacy Data Import

Legacy data refers to data applicable to Oracle Student System that currently resides in another system or data-source and needs to be imported.

The legacy import interfaces allow institutions to import legacy data in bulk into the following subsystems of Oracle Student System:

- Program Structure and Planning
- Enrollments
- Academic Records

Enrollments and Academic Records use a common import interface while Program Structure and Planning has its own interface for import.

Legacy data import process is described in detail in the following chapter:

Legacy Data Import Processes, page 2-1.

Admissions Module

This section describes how to import person and admission information into the Admissions module of Oracle Student System.

Admissions Import Process

The Admissions Import process is a concurrent program that is used to collect and transfer the following types of person and admission information from interface tables to Oracle Student System target tables:

- person information
- admission application information
- test results
- transcripts

Details on the Admissions Import process are provided in the following chapter:

- Admissions Import Process, page 3-1

Institutions Import Process

This chapter describes the process of importing data about institutions into Oracle Student System.

Details on the Institutions Import process are provided in the following chapter:

- Institutions Import Process, page 4-1

Probability Values Import Process

The Probability Values Import process is a concurrent program that enables users to collect and transfer probability values from the interface tables in the Admissions module to the Oracle Student System target table.

Details on the Probability Values Import process are provided in the following chapter:

- Probability Values Import Process, page 5-1

Academic Index Import Process

The Academic Index Import process is a concurrent program that enables users to collect and transfer index data from interface tables to Oracle Student System target tables in the Admissions module.

Details on the Academic Index Import process are provided in the following chapter:

- Academic Index Import Process, page 6-1

Test Score Conversion and Import Process

The Test Score Conversion and Import process enables users to convert and transfer test scores individually or as a batch to interface tables, and then run a concurrent program to collect and transfer the test scores from the interface tables to Oracle Student System target tables.

Details on the Test Score Conversion and Import process are provided in the following chapter:

- Test Score Conversion and Import Process, page 7-1

Automatic Batch Admission Decisions Import Process

The Automatic Batch Admission Decisions Import process is an internal process that enables users to collect and transfer batch admission decisions from interface tables to Oracle Student System target tables.

Details on the Automatic Batch Admission Decisions Import process are provided in the following chapter:

- Automatic Batch Admission Decisions Import Process, page 8-1

Enrollments Module

This section describes how to upload enrollment registration information and how to perform enrollment functions using interactive voice response (IVR) systems.

Enrollment Registration Update Process

Institutions in the UK require a mechanism to record that a student has actually completed registration. The date of registration must be captured and if the student is pre-enrolled in any units, the unit attempts must be confirmed. The enrollment registration update process enables the institutions update registration information and confirm pre-enrolled units.

Enrollment registration update process is described in detail in the following chapter:

Enrollment Registration Update Process, page 10-1.

Interactive Voice Response Interface

The Interactive Voice Response (IVR) APIs allow institutions that have IVR systems to communicate with Oracle Student System and perform enrollment functions. Oracle Student System provides these APIs which function as an interface between the institution’s IVR and Oracle Student System database.

The IVR systems provided by the institutions send data such as the student’s login to Oracle Student System and retrieve information which needs to be given to the student. This is carried out with the help of the IVR APIs.

IVR processes are described in detail in the following chapter:

Interactive Voice Response Process, page 11-7.

Financial Aid Module

This section describes how to upload work-study payroll data from an internal or external Human Resources or Payroll system to the target table in Oracle Student System.

Student Employment Payroll Data Upload Process

The Upload Payment Details concurrent process enables users to upload payroll data from the Student Employment Payment Details Interface table to the Student Employment Payment Details target table in Oracle Student System for tracking student employee earnings related to work-study awards.

Details of Upload Payment Details process are provided in the following chapter:

- Student Employment Payroll Data Upload Process, page 12-1

Student Finance Module

This section describes how to transfer student financial data within the Student Finance module of Oracle Student System.

External Charges Import Process

The External Charges Import process is a concurrent program that enables users to collect and transfer external charges from the External Charges Interface table to the Import Charges target table in the Student Finance module.

Details on the External Charges Import process are provided in the following chapter:

- External Charges Import Process, page 13-1

Import and Calculate Ancillary Charges Process

The Import and Calculate Ancillary Charges process is a concurrent program that enables users to collect ancillary attributes from the Ancillary Attributes Interface table, calculate the ancillary charges for those attributes, and transfer the calculated charges to the Import Charges target table in the Student Finance module.

Details on the Import and Calculate Ancillary Charges process are provided in the following chapter:

- Import and Calculate Ancillary Charges Process, page 14-1

Credits API

The Credits API is a public API that can be used internally or externally to create payment or credit information in the Student Finance module of Oracle Student System.

Details on the Credits API are provided in the following chapter:

- Credits API, page 15-1

Student Refund Transfer Process

The Student Refund Transfer process comprises one concurrent program provided by Oracle Student System and one user-defined transfer process using third party tools as follows:

- The Transfer Refunds Information to Refunds Interface process collects and transfers student refunds from the Refunds table in the Student Finance module to the Refunds Interface target table.
- Users employ third party tools to transfer student refund data from the Refunds Interface target table to Payables for payment processing.

Details on the Student Refund Transfer process are provided in the following chapter:

- Student Refund Transfer Process, page 16-1

Posting Interface Process

The Posting Interface process is a concurrent program that enables users to transfer transactions with accounting implications from three areas in the Student Finance module of Oracle Student System to the Posting Interface target table in the Student Finance module.

Details on the Posting Interface process are provided in the following chapter:

- Posting Interface Process, page 17-1

Sponsorship Assignment API

The Sponsorship Assignment API provides a means to create multiple student-sponsor assignment records without manual entry.

Details on the Sponsorship Assignment API are provided in the following chapter:

- Sponsorship Assignment API, page 18-9

Program Structure and Planning Module

This section describes how to transfer Unit Section Occurrence information from Oracle Student System to a third party scheduling application and back to Oracle Student System.

Scheduling Interface Process

This chapter describes the process of transferring data between the Oracle Student System database and third party scheduling software. This process is usually completed by staff who schedule courses or Unit Section Occurrences.

Details on the Scheduling Interface process are provided in the following chapter:

- Scheduling Interface Process, page 19-1

Academic Records Module

This section describes how to transfer data from Oracle Student System to a third party degree audit application and back to Oracle Student System.

Degree Audit Data Exchange Process

The Degree Audit Outgoing Data process enables users to transfer information to be used for a degree audit from Oracle Student System to outgoing interface tables. This data then flows from the outgoing interface tables to a third party degree audit application.

The Degree Audit Incoming Data process enables users to transfer degree audit information from a third party degree audit application to incoming interface tables.

Details on the Degree Audit Outgoing and Incoming Data processes are provided in the following chapter:

- Degree Audit Data Exchange Process, page 20-1

HESA Data Import

Institutions in the UK must report student data to Higher Education Statistics Agency (HESA). Institutions can use HESA information of new students provided by Universities and Colleges Administration Service (UCAS) for reporting to HESA. The information provided in a fixed-format file is imported into Oracle Student System through the HESA data import process.

The HESA data import process is described in detail in the following chapter:

HESA Data Import Process, page 21-1.

References

This section describes technical reference information.

Admission Import Process Error Codes

Institutions must verify the validity of data elements before importing data into Oracle Student System to avoid errors. Error codes are used to update the interface table affected by the error condition. An error report is generated indicating why the import process was interrupted.

Details on the import process error codes are provided in the following appendix:

- Admission Import Process Error Codes, page A-1

Admission Import Source Category Tables

When users select a data source type to import into the Admissions module, specific source categories are imported for the system source type selected for import. Some source categories must be imported, while others are optional.

Details on the system source types and their associated source categories and tables are provided in the following appendix:

- Admission Import Source Category Tables, page B-1

Part I

Legacy Data Import

2

Legacy Data Import Processes

This chapter describes the process of importing legacy data pertaining to program structure and planning, records, and enrollments into Oracle Student System tables. The following sections are in this chapter:

Definition

Legacy data refers to data applicable to Oracle Student System that currently resides in another system or data-source and needs to be imported.

The legacy import interfaces allow institutions to import legacy data in bulk into the following subsystems of Oracle Student System:

- Program Structure and Planning

- Enrollments
- Academic Records

Enrollments and Academic Records use a common import interface while Program Structure and Planning has its own interface for import.

Overview

Oracle Student System provides interface tables which facilitate the import of legacy data in a specified format. The institutions are required to load the interface tables with legacy data that they wish to import into Oracle Student System.

Institutions must map the legacy data to Oracle Student System data structures before uploading the data to the interface tables. Data can be loaded into interface tables manually or using automated data entry tools.

After the data is loaded into interface tables, a concurrent process can be run that calls several individual APIs to validate the data, and on successful validation, writes the data into Oracle Student System tables. The APIs are listed under Program Structure and Planning Legacy Data Import Process and Enrollments and Records Legacy Data Import Process.

If the validation fails, and an error is reported, you must resolve the error in the interface table and change the status to R (Ready for reprocessing). The record is taken up for import into Oracle Student System in the next run of the concurrent process.

Program Structure and Planning contains definitions of programs, unit sets, and units and their associated offering structures. This information must be entered into various setup windows before any related legacy data can be imported.

For information on the prerequisites for importing legacy data, see Minimum Setup Requirements for Legacy Data in *Oracle Student System Implementation Guide*.

In the Program Structure and Planning legacy import interface, the following items are supported:

- Unit versions and associated information required for an active, inactive, or planned version including teaching responsibilities, disciplines, grading schemas, mandatory reference codes, and subtitles
- Unit sections and associated mandatory information including credit points, grading schemas, reference codes, and occurrences

The validations performed on the data being imported are listed in the section Validations on Program Structure and Planning Legacy Data.

In enrollments legacy data import, students are brought in with the status appropriate to the units and other attributes of their enrollment data.

For instance, if the program is active and the student has units, then they are brought in as enrolled. If a current intermission is imported, then the student is imported as Intermitted. If the program has been completed, then the status is completed, etc.

Fee assessment is not kicked off by the legacy import process. Fee assessment must be run manually after successful import of the enrollment legacy data.

Importing of the following data is supported:

- Program Attempts (& Research Candidatures)
- Unit Set Attempts
- Intermissions
- Award Aims
- Research Supervisors
- Research Thesis & Examinations
- Student Unit Attempts
- Program HESA Statistics
- Unit Set (YoP) HESA Details
- Graduand details
- Level Advanced Standing details
- Unit Advanced Standing details
- Unit Outcome details

- Progression details
- Class Rank details

The validations performed on the data being imported are listed in the section Validations on Enrollments and Records Legacy Data.

Program Structure and Planning Legacy Data Import Process

Before importing legacy data, you must set up Program Structure and Planning, Calendar, Teaching calendars, and Programs and Unit Sets. For information on the prerequisites for importing legacy data, see Minimum Setup Requirements for Legacy Data in *Oracle Student System Implementation Guide*.

The Program Structure and Planning Legacy Data Import process is outlined below.

1. Institutions enter data into interface tables provided by Oracle Student System.

   - Data is loaded in batches, each requiring a row placed in the IGS_PS_LGCY_BAT_INT table, with an appropriate BATCH_ID and description information.
   - Rows to be imported within the batch are placed in their corresponding interface table, along with the matching BATCH_ID and a unique row ID in the second …_ID column (e.g., UNIT_VERSION_INT_ID in the unit version interface table IGS_PS_LGCY_UV_INT). New rows must be inserted with a record status of U (unprocessed).

   These interface tables are listed under Program Structure and Planning Legacy Data Import Interface Tables. Further information on the required columns can be found in the *Oracle eBusiness Suite Technical Reference Manual*.

   Users are responsible for mapping legacy data to Oracle Student System interface table columns and then entering data into them.

2. You can then run the Legacy Import Process for Program Structure and Planning concurrent process, indicating the batch ID of the rows to be imported. The list of batches available for import is shown based on the contents of the IGS_PS_LGCY_BAT_INT table.

   The import process includes the following sub processes, corresponding to each interface table: Import Unit Version - Teaching Responsibility - Unit Discipline - Unit Grading Schema - Unit Section - Unit Section Grading Schema - Unit Section Occurrences - Unit Reference Codes

The rows in these tables are processed one unit version at a time, and on passing all validations are committed before continuing to the next unit version within the batch.

3. The concurrent process rolls back for a specific unit if any of the sub processes fail at any stage. If validation fails for a particular set of data, the process takes up the subsequent data set.

   Each API validates the parameters, data integrity and business logic against the Oracle Student System's legacy data rules as detailed under Validations on Program Structure and Planning Legacy Data.

   If the record already exists in Oracle Student System, it is not updated or overwritten by the record in the interface table.

4. Rows that are successfully imported will have their status changed to I (imported), or they will be deleted depending on the parameter settings.

   Any errors/warnings are saved in the interface table IGS_PS_LGCY_ERR_INT. Rows that weren't loaded due to errors will have their status changed in the interface table to E (error). Rows that were loaded with warnings will have their record status changed to W (warning).

   All errors, warnings, and successful imports are also written into the concurrent manager log. These are referenced by their row ID, as specified in the interface table.

   You can resolve the errors by updating the interface table data as appropriate to correct the data and resetting the status of the record in the interface table to R (ready for reprocessing).

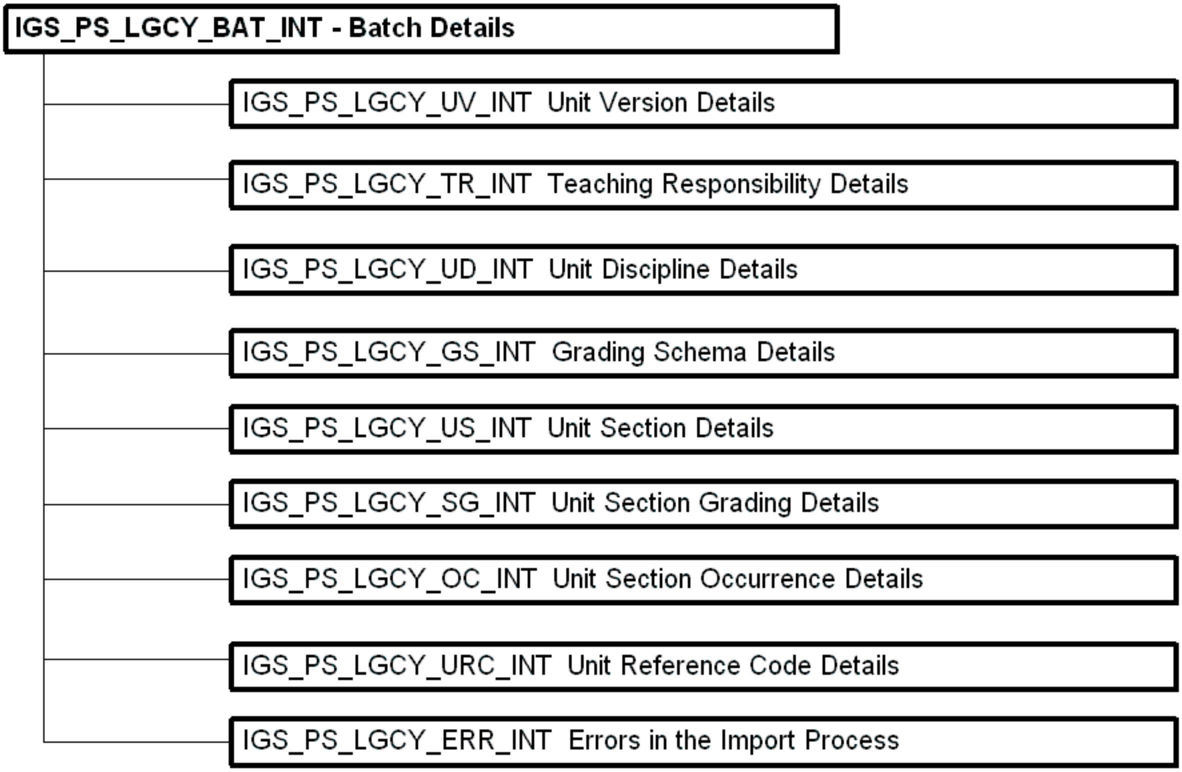
   The process takes only records with status R (ready for reprocessing) and U (unprocessed) for import.

   For information on running the legacy import process concurrent process, see Chapter 2, Program Structure and Planning, *Oracle Student System User's Guide*.

Program Structure and Planning Legacy Import Table Structure

Figure 2–1 shows the master table, IGS_PS_LGCY_BAT_INT and the various child tables associated with it.

Figure 2–1 Program Structure and Planning Legacy Data Interface Table Structure


IGS_PS_LGCY_BAT_INT - Batch Details
IGS_PS_LGCY_UV_INT Unit Version Details
IGS_PS_LGCY_TR_INT Teaching Responsibility Details
IGS_PS_LGCY_UD_INT Unit Discipline Details
IGS_PS_LGCY_GS_INT Grading Schema Details
IGS_PS_LGCY_US_INT Unit Section Details
IGS_PS_LGCY_SG_INT Unit Section Grading Details
IGS_PS_LGCY_OC_INT Unit Section Occurrence Details
IGS_PS_LGCY_URC_INT Unit Reference Code Details
IGS_PS_LGCY_ERR_INT Errors in the Import Process

Validations on Program Structure and Planning Legacy Data

The following tables list the validations performed at the time of importing data into Oracle Student System. It would help you to prepare the data before entering it into the interface table.

Table 2–1 lists the validations performed for creating unit version records. This information is useful for entering data into the IGS_PS_LGCY_UV_INT table.

Table 2–1 Validations performed for creating unit version records

Validation	Validation Result (Error/Warning)
Unit version must not already exist.	Warning
Version end date must be greater than or equal to the version start date.	Error
Version end date must be greater than or equal to the version expiration date.	Error
Version expiration date must be greater than or equal to the version start date.	Error
Cannot set end date of a unit version, which is not having an inactive status.	Error
Cannot set status of a unit version to inactive with an open end date.	Error
The unit coordinator must be a staff member.	Error
IF the unit version has the assessable indicator selected, then it must have the supplementary exams selected.	Error
At least one unit enrollment method type between Interactive Voice Response and Self Service must be selected.	Error
Only one version of unit can exist with active status and expiry date not set.	Error
A unit version's status cannot be inactive if it has active or planned subordinate units.	Error
A unit version's status cannot be active if it has inactive or planned superior units.	Error
A unit version's status cannot be active if it has inactive or planned subordinate units.	Error
If the points override indicator is set, then the minimum, maximum and increment points field as required.	Error
Approval date must be earlier than the start date.	Error
The Override Credit Point Maximum must be greater than or equal to the Override Credit Point Minimum.	Error
The Enrolled credit points must be greater than or equal to the Override Credit Point Minimum.	Error

Table 2–1 Validations performed for creating unit version records

Validation	Validation Result (Error/Warning)
The Override credit points Maximum must be greater than or equal to the Enrolled credit points.	Error
The Achievable credit points must be greater than or equal to the Override Credit Point Minimum.	Error
The Override credit points Maximum must be greater than or equal to the Achievable credit points.	Error
The Credit Point Minimum must be a decrement (as per the Credit Point Increment value) of Enrolled credit points.	Error
The Credit Point Maximum must be an increment (as per the Credit Point Increment value) of Enrolled credit points.	Error
The override credit point increment, override credit point maximum and override credit point minimum fields can only be set if the override credit points indicator has been secluded.	Error
The subtitle ID must not already exist.	Warning
If there exist approved subtitle(s) for a unit version and any other sub-title is entered then it throws the error message otherwise allows to enter any free text as sub title if as unapproved.	Error
Curriculum ID must not already exist.	Warning
If re-enrollment for credit (Repeatable_Ind) is not allowed, then max_repeats_for_credit, max_repeat_credit_points, and max_repeats_for_funding must be null.	Error
LEVEL_CODE is mandatory for creating unit version records.	Error
AUDIT_PERMISSION_IND can have value ’Y’ only if AUDITABLE_IND has value ‘Y’.	Error
MAX_AUDITORS_ALLOWED can have value only if AUDITABLE_IND has value ‘Y’.	Error

Table 2–2 lists the validations performed for creating teaching responsibility records. This information is useful for entering data into the IGS_PS_LGCY_TR_INT table.

Table 2–2 Validations performed for creating teaching responsibility records

Validations	Validation Result (Warning/Error)
Teaching Responsibility records that add up to 100% must not already exist.	Warning
The percentage allocation for teaching responsibility must add up to 100 for a unit version.	Error

Table 2–3 lists the validations performed for creating unit discipline records. This information is useful for entering data into the IGS_PS_LGCY_UD_INT table.

Table 2–3 Validations performed for creating unit discipline records

Validations	Validation Result (Warning/Error)
Unit discipline records that add up to 100% must not already exist.	Warning
The percentage allocation for unit discipline must add up to 100.	Error

Table 2–4 lists the validations performed for creating grading schema records. This information is useful for entering data into the IGS_PS_LGCY_GS_INT table.

Table 2–4 Validations performed for creating grading schema records

Validations	Validation Result (Warning/Error)
Unit version grading schema must not already exist.	Warning
Grading schema entered must have a grading schema of type Unit Grading Schema.	Error
One and only one unit grading schema code can be marked as the default for a unit version.	Error

Table 2–5 lists the validations performed for creating unit section records. This information is useful for entering data into the IGS_PS_LGCY_US_INT table.

Table 2–5 Validations performed for creating unit section records

Validation	Validation Result (Error/Warning)
Unit section must not already exist.	Warning
If the Waitlist Allowed check box is selected, then maximum student per waitlist can be greater than zero. If waitlist is not allowed, maximum students per waitlist cannot be specified.	Warning
If organization level waitlist is not allowed, then waitlist is not allowed at unit offering pattern level as well.	Error
Unit contact for unit section must be a staff member.	Error
The location type of the location code used in a unit section must be of type CAMPUS.	Error
Unit enrollment method type must be Interactive Voice Response or Self Service.	Defaulted from unit version
Call number must be unique across a load calendar.	Error
Anonymous unit grading must be enabled at the unit level before it can be enabled at the unit section level.	Error
If configuration of the anonymous grading method is not done, then the Anonymous Unit Grading check box cannot be selected.	Error
Anonymous assessment grading must be enabled at the unit level before it can be enabled at the unit section level.	Error
If the configuration of the anonymous grading method is not done, then the Anonymous Assessment Grading check box cannot be selected.	Error
Grading schema code and version number values to be fetched from unit level before inserting.	Defaulted from unit level if NULL
Anonymous assessment grading, anonymous unit grading to be defaulted from unit level if NULL.	Defaulted from unit level if NULL
Unit section credit points must not already exist.	Warning
If points override indicator is selected, then minimum and maximum credits points are mandatory.	Error
The credit point maximum must be greater than or equal to the credit point minimum.	Error
The enrolled credit points must be greater than or equal to the credit point minimum.	Error
The credit points maximum must be greater than or equal to the enrolled credit points.	Error

Table 2–5 Validations performed for creating unit section records

Validation	Validation Result (Error/Warning)
The achievable credit points must be greater than or equal to the credit point minimum.	Error
AUDIT_PERMISSION_IND can have value ’Y’ only if AUDITABLE_IND has value ‘Y’	Error
Call Number should not have value if the profile ‘IGS: Call Number For Unit Section’ is set to None.	Error
More than one to-be-announced unit section occurrence must not exist for a Unit Section.	Error
More than one unit section occurrence must not exist with given information.	Error

Table 2–6 lists the validations performed for creating unit section records. This information is useful for entering data into the IGS_PS_LGCY_SG_INT table.

Table 2–6 Validations performed for creating unit section grading records

Validations	Validation Result (Warning/Error)
Unit section grading schema record must not already exist.	Warning
For a unit section, at least and at most one record must have default column value as 'Y'.	Error
Grading schema code must have the correct type for the unit section level.	Error

Table 2–7 lists the validations performed for creating unit section occurrence records. This information is useful for entering data into the IGS_PS_LGCY_OC_INT table.

Table 2–7 Validations performed for creating unit section occurrence records

Validation	Validation Result (Error/Warning)
Start date cannot be earlier than unit section start date.	Error
Start date cannot be later than unit section end date.	Error
Start date cannot be later than end date.	Error
End date cannot be later than unit section end date.	Error

Table 2–7 Validations performed for creating unit section occurrence records

Validation	Validation Result (Error/Warning)
End date cannot be earlier than unit section start date.	Error
Start time cannot be later than end time.	Error
If occurrence is not TBA, then at least one day must be selected.	Error
You can only set either preferred or dedicated building / room values.	Error
If scheduled room / building are set, then neither preferred nor dedicated room values can be set.	Error

Table 2–8 lists the validations performed for creating unit reference code records. This information is useful for entering data into the IGS_PS_LGCY_URC_INT table.

Table 2–8 Validations performed for creating unit reference code records

Validations	Validation Result (Warning/Error)
If data type is UNIT, then unit reference code must not already exist.	Warning
If data type is SECTION, then unit section reference code must not already exist.	Warning
If data type is OCCURRENCE, then unit section occurrence reference code must not already exist.	Warning
If data type is not UNIT, then UNIT_CD, VERSION_NUMBER, TEACH_CAL_ALTERNATE_CODE, LOCATION_CD, UNIT_CLASS must be specified.	Error
If data type is OCCURRENCE, then START_TIME, END_TIME must be specified.	Error
If data type is OCCURRENCE, then one of: MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY must be specified.	Error

Program Structure and Planning Legacy Data Import Interface Tables

Institutions must enter data in the following interface tables before the Import process.

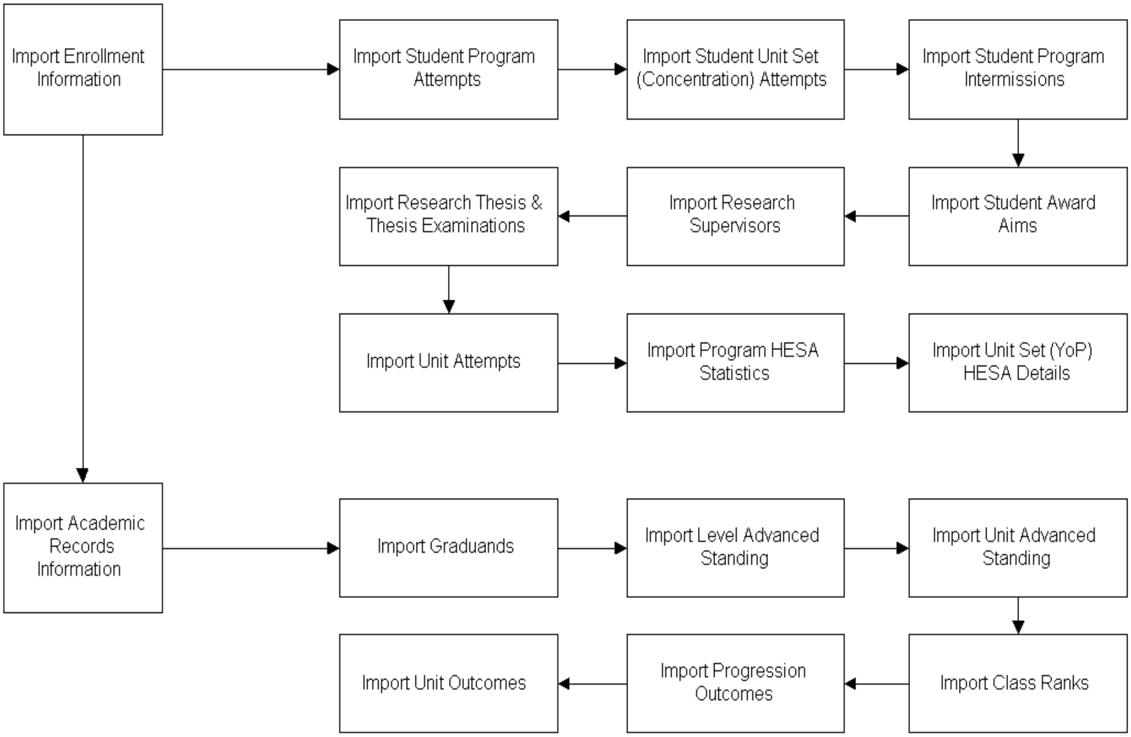
- IGS_PS_LGCY_BAT_INT
- IGS_PS_LGCY_ERR_INT
- IGS_PS_LGCY_UV_INT
- IGS_PS_LGCY_TR_INT
- IGS_PS_LGCY_UD_INT
- IGS_PS_LGCY_GS_INT
- IGS_PS_LGCY_US_INT
- IGS_PS_LGCY_SG_INT
- IGS_PS_LGCY_OC_INT
- IGS_PS_LGCY_UR_INT

For information on table column descriptions and values, see Oracle Student System Table Information at http://metalink.oracle.com/metalink/plsql/ml2_gui.startup.

Enrollments and Records Legacy Data Import Process Functional Flowchart

Figure 2–2 shows the Enrollments Legacy Data Import process from a functional perspective.

Figure 2–2 Enrollments Legacy Data Import Process Flow


Import Enrollment Information
Import Student Program Attempts
Import Student Unit Set (Concentration) Attempts
Import Student Program Intermissions
Import Student Award Aims
Import Research Supervisors
Import Research Thesis & Thesis Examinations
Import Unit Attempts
Import Program HESA Statistics
Import Unit Set (YoP) HESA Details
Import Academic Records Information
Import Graduands
Import Level Advanced Standing
Import Unit Advanced Standing
Import Class Ranks
Import Progression Outcomes
Import Unit Outcomes

Enrollments and Records Legacy Data Import Process

Before importing legacy data, you must set up some elements of the system. For information on the prerequisites for importing legacy data, see Minimum Setup Requirements for Legacy Data in *Oracle Student System Implementation Guide*.

The Enrollments and Records Legacy Data Import process is outlined below.

1. Institutions enter data into interface tables provided by Oracle Student System.

   - Data is loaded in batches, each requiring a row placed in the IGS_EN_LGCY_BAT_INT table, with an appropriate BATCH_ID and description information.

   - Rows to be imported within the batch are placed in their corresponding interface table, along with the matching BATCH_ID and a unique row ID in the second …_ID column (e.g., LEGACY_EN_SPA_INT_ID in the student program attempt interface table IGS_EN_LGCY_SPA_INT). New rows must be inserted with a record status of U (unprocessed).

   These interface tables are listed under Enrollments and Records Legacy Data Interface Tables. Further information on the required columns can be found in the *Oracle eBusiness Suite Technical Reference Manual*.

   Users are responsible for mapping legacy data to Oracle Student System interface table columns and then entering data into them.

2. You can then run the Legacy Import Process for Enrollments and Records concurrent process, indicating the batch ID of the rows to be imported. The list of batches available for import is shown based on the contents of the IGS_EN_LGCY_BAT_INT table.

3. There are nine enrollments-related interfaces, and six records-related interfaces from which all rows in the batch will be processed in the order outlined in the process flow diagram above. Once validations are passed, each row will be committed to the database before continuing to the next.

   Each API validates the parameters, data integrity and business logic against the Oracle Student System’s legacy data rules as detailed in this document.

   If the record already exists in Oracle Student System, it is not updated or overwritten by the record in the interface table.

4. Rows that are successfully imported will have their status changed to I (imported), or they are deleted depending on the parameter settings.

Any errors/warnings are saved in the interface table IGS_EN_LGCY_ERR_INT. Rows that weren’t loaded due to errors will have their status changed in the interface table to E (error). Rows that were loaded with warnings will have their record status changed to W (warning).

All errors, warnings, and successful imports are also written into the concurrent manager log. These are referenced by their row ID, as specified in the interface table.

You can resolve the errors by updating the interface table data as appropriate to correct the data and resetting the status of the record in the interface table to R (ready for reprocessing).

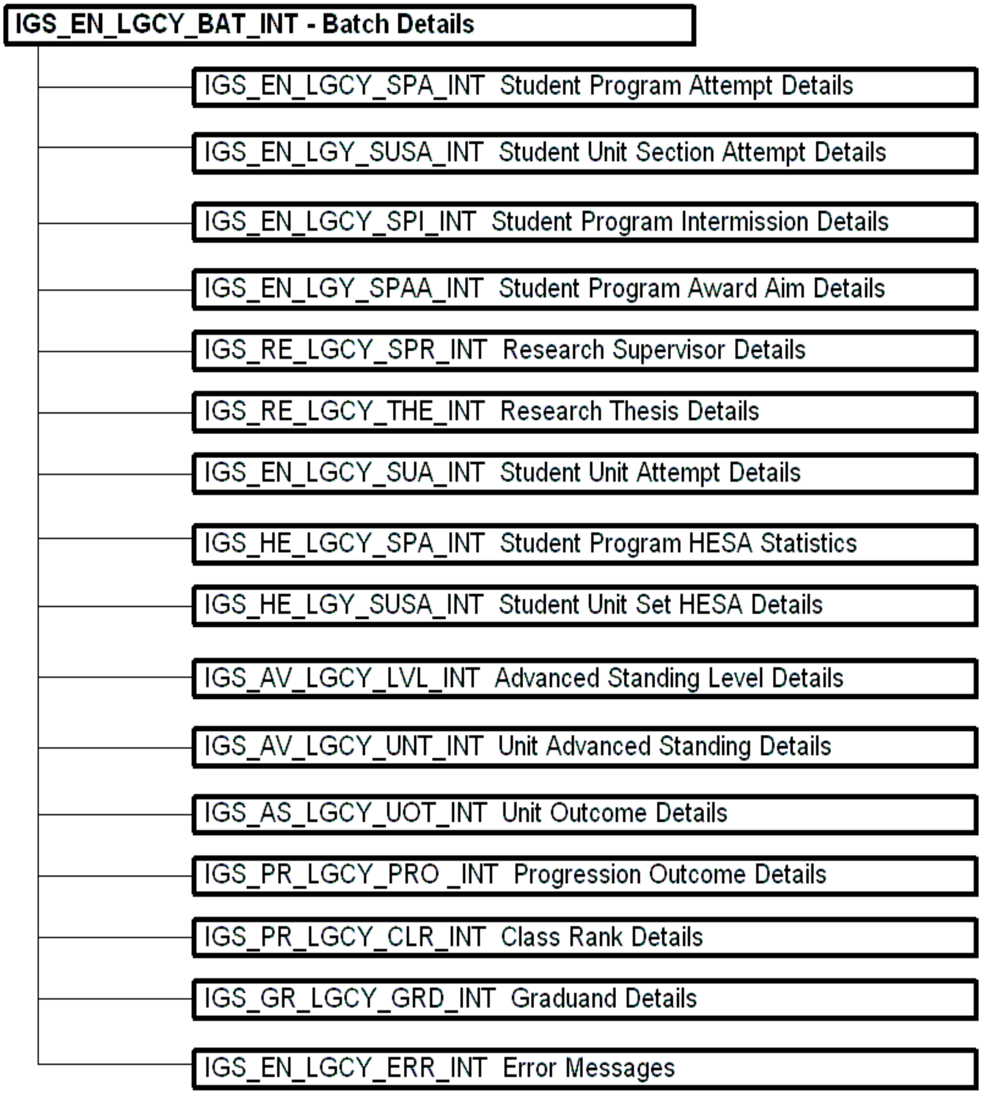
The process only takes records with status U (unprocessed) and R (ready for reprocessing) for processing.

For information on running the legacy import process concurrent process, see Chapter 5, Enrollments, *Oracle Student System User’s Guide*.

Enrollments and Records Legacy Import Table Structure

Figure 2–3 shows the master table, IGS_EN_LGCY_BAT_INT and the various child tables associated with it.

Figure 2–3 Enrollment Legacy Data Interface Table Structure


IGS_EN_LGCY_BAT_INT - Batch Details
IGS_EN_LGCY_SPA_INT Student Program Attempt Details
IGS_EN_LGY_SUSA_INT Student Unit Section Attempt Details
IGS_EN_LGCY_SPI_INT Student Program Intermission Details
IGS_EN_LGY_SPAA_INT Student Program Award Aim Details
IGS_RE_LGCY_SPR_INT Research Supervisor Details
IGS_RE_LGCY_THE_INT Research Thesis Details
IGS_EN_LGCY_SUA_INT Student Unit Attempt Details
IGS_HE_LGCY_SPA_INT Student Program HESA Statistics
IGS_HE_LGY_SUSA_INT Student Unit Set HESA Details
IGS_AV_LGCY_LVL_INT Advanced Standing Level Details
IGS_AV_LGCY_UNT_INT Unit Advanced Standing Details
IGS_AS_LGCY_UOT_INT Unit Outcome Details
IGS_PR_LGCY_PRO _INT Progression Outcome Details
IGS_PR_LGCY_CLR_INT Class Rank Details
IGS_GR_LGCY_GRD_INT Graduand Details
IGS_EN_LGCY_ERR_INT Error Messages

Validations on Enrollments and Records Legacy Data

The following tables list the validations performed at the time of importing data into Oracle Student System. It would help you to prepare the data before entering it into the interface tables.

Table 2–9 lists the validations performed while creating student program attempt records. This information is useful for entering data into the IGS_EN_LGCY_SPA_INT table.

Note: Make sure the column PRIMARY_PROG_TYPE_SOURCE contains upper case "SYSTEM" for the record being imported.

Table 2–9 Validations for creating student program attempt records

Validation	Validation Result (Error/Warning)
If attempt is confirmed, then commencement (start) date must be set (and vice versa).	Error
Discontinuation date cannot be set when program attempt is not confirmed, or requirements complete flag is set.	Error
Discontinuation date must be later than the commencement (start) date.	Error
Discontinuation date cannot be a future date.	Error
Discontinuation reason code can only be set in conjunction with discontinuation date.	Error
If discontinuation date is set, then discontinuation reason code must also be set.	Error
Lapsed date can only be set where program attempt is confirmed and not completed.	Error
Cannot set requirements complete flag when program attempt is unconfirmed.	Error
Requirements complete date cannot be a future date.	Error
Requirements complete date cannot be prior to commencement (start) date.	Error
Requirements complete date can only, and must be set when the requirements complete flag is set.	Error
Only one program per career (program type) can be marked as Primary (career-centric only).	Error
Only one program can be flagged as the key program for a student.	Error
There must always be one program flagged as the key program for a student.	Error
Key program must also be a primary program (career-centric Only).	Error
Primary program type must be set (career-centric only).	Error

Table 2–9   Validations for creating student program attempt records

Validation	Validation Result (Error/Warning)
Student program attempt must not already have a candidature.	Warning
If not specified, minimum submission date must be derivable.	Error
Minimum submission date cannot be earlier than program attempt commencement (start) date.	Error
Minimum submission date cannot be later than the maximum submission date.	Error
If not specified, maximum submission date must be derivable.	Error
Maximum submission date cannot be before the minimum submission date.	Error
Research topic is required when program attempt is confirmed.	Error

Table 2–10 lists the validations performed while creating student unit set attempt records. This information is useful for entering data into the IGS_EN_LGY_SUSA_INT table.

Table 2–10   Validations for creating student unit set attempt records

Validation	Validation Result (Error/Warning)
Student unit set attempt must not already exist. If the same same person number, program code, unit set code, version number, or selection date exists, the record is considered a duplicate.	Warning
Unit set version must be offered within the students’ program offering option.	Error
Record cannot be created for an attempt of a unit set that has already been completed by the student.	Error
Records cannot be created for two active attempts of the same unit set within a single program attempt.	Error
The voluntary end indicator can only be set when the attempt end date is set.	Error
An administrative unit set cannot be specified as the primary set.	Error
If the authorized date is set, then the authorized person must also be set (and vice versa).	Error
The authorized date/person can only be set if the unit set is being ended, or if the unit set version is flagged in Program Structure and Planning (PSP) as requiring authorization to enroll.	Error

Table 2–10 Validations for creating student unit set attempt records

Validation	Validation Result (Error/Warning)
Authorizing person must be a staff member.	Error
If unit set is flagged (in PSP) as requiring authorization, then the authorized person/date must be set.	Error
If unit set was part of admissions offer, and end date is set, then authorization date/person must be set.	Error
If the requirements complete date is set, then the requirements complete flag must also be set.	Error
The completion flag/date can only be set if the unit set attempt has been confirmed.	Error
Completion source type can only be set if the completion flag is set.	Error
The selection date must be set if the confirmed flag is set (and vice versa).	Error
The confirmed indicator cannot be set if the end date or completion date is set.	Error
The confirmed indicator cannot be set if the program attempt is unconfirmed.	Error
If the unit set is specified within PSP as a child, then it must be a child attempt.	Error
If the unit set has a parent, then the relationship must be valid against those specified in PSP.	Error
A unit set attempt cannot be a direct or indirect parent of itself.	Error
The parent unit set attempt must be within the same program attempt as the child.	Error
Record cannot be created for a confirmed parent if the attempt is not also confirmed.	Error
If parent unit set is ended, then the attempt must also be ended.	Error
The confirmed indicator cannot be set if parent unit set is not also confirmed.	Error
Unit sets with category of pre-enrollment year cannot be created unless the profile option is set.	Error
Unit set attempts of set with category of pre-enrollment year cannot overlap selection/completion dates.	Error

Table 2–11 lists the validations performed while creating student unit attempt records. This information is useful for entering data into the IGS_EN_LGCY_SUA_INT table.

Table 2–11 Validations performed for creating student unit attempt records

Validation	Validation Result (Warning/Error)
The student unit attempt must not already exist. Records with the same person number, program code, unit code, or teaching calendar are considered duplicates.	Warning
Unit version must not be planned.	Error
Unit section must have offered flag set.	Error
If enrolled date is set, program attempt must be confirmed.	Error
Enrolling into a teaching period within period of intermission - defined as intermission dates overlapping census dates - is not allowed.	Error
Unit version cannot be enrolled if unit version is not repeatable and advanced standing has been already granted for the unit.	Error
Alternative subtitle is only permitted if unit version title override indicator is set.	Error
Override achievable or enrolled credit points are only permitted if unit version points override indicator is set.	Error
Override achievable credit points must be within limits set by unit version/section minimum, maximum, and increment values.	Error
Override enrolled credit points must be within limits set by unit version/section minimum, maximum, and increment values	Error
Grading schema code/version must be valid within the enrolling unit section/version.	Error
If enrolled date is set, then program attempt status cannot be Unconfirmed.	Error
If program attempt is discontinued, then the teaching period start date cannot be after the discontinued date of the program.	Error
If career centric, and program is a secondary (i.e., non-primary) program then enrolled units cannot be attached.	Error
If unit attempt is not unconfirmed or duplicate, then the enrolled date cannot be prior to the program commencement (start) date.	Error
Enrolled unit attempts cannot be added within a discontinued or completed program.	Error

Table 2–11 Validations performed for creating student unit attempt records

Validation	Validation Result (Warning/Error)
No assessment indicator (i.e., audit flag) can only be set when unit section permits auditing.	Error
Research unit must have teaching period covered by research supervision.	Error
Research candidature must exist in order to enroll in research unit.	Error
Discontinued date cannot be a future date.	Error
Discontinued date cannot be prior to unit attempt enrolled date.	Error
Discontinued date must be set if administrative unit status is set.	Error
Administrative unit status can only be set if discontinued date is set.	Error
Administrative unit status must be set if discontinued date is set.	Error
If administrative unit status is set, then it must be possible to determine applicable grade from setup.	Error
Discontinuation reason code must match a row with the unit flag set.	Error
If dropped indicator is set, then enrolled date must be set.	Error
If dropped indicator is set, then discontinued date must not be set.	Error
Discontinuation reason code can only be set when discontinued date is set.	Error
Transfer program code cannot be the same as the program of the unit attempt being imported.	Error
If any of the transfer columns are specified, then another unit attempt must exist with matching person ID, transfer program code, unit code, teaching alternate code (resolved to calendar type and sequence number), location code and unit class. At least one row must exist. If multiple rows exist, then location or class indicator must be selected. Status must be COMPLETED or DISCONTIN.	Error
If transfer date is specified, then ENROLLED_DT, DISCONTINUED_DT, ADMINISTRATIVE_UNIT_STATUS, DCNT_REASON_CD, NO_ASSESSMENT_IND, OVERRIDE_ENROLLED_CP, OVERRIDE_ACHIEVABLE_CP, GRADING_SCHEMA_CD, GS_VERSION_NUMBER, SUBTITLE, OUTCOME_DT, MARK, GRADE cannot be specified.	Error
If career centric, and unit transfer, and transferring between programs of different program types (i.e., between careers), then source program attempt must have discontinuation reason code of type TRANSFER.	Error
If program centric, and unit transfer, then source program attempt must have discontinuation reason code of type TRANSFER.	Error

Table 2–11 Validations performed for creating student unit attempt records

Validation	Validation Result (Warning/Error)
Grade can only be set when outcome date is set, and vice versa.	Error
Mark can only be set when outcome date is set.	Error
If grade set and outcome grading schema code and version are not set, then grade must exist in the unit section default grading schema.	Error
If outcome date set, then enrolled date must be set and discontinued must not be set.	Error
If adding completed unit attempt to a discontinued program attempt, the outcome date cannot be after the discontinuation date.	Error
If waitlist date is set, then enrolled date and discontinued date must not be set.	Error

Table 2–12 lists the validations performed while creating student program intermission records. This information is useful for entering data into the IGS_EN_LGCY_SPI_INT table.

Table 2–12 Validations performed for creating student program intermission records

Validations	Validation Result (Warning/Error)
Intermission record must not already exist with the same start and date dates.	Warning
Start date must not be later than end date.	Error
Periods of intermission within the same program cannot overlap.	Error
Period of intermission cannot overlap enrolled / completed unit attempt teaching period census dates.	Error
Approved person only available when intermission type flagged as requiring approval	Error
Institution name, maximum credit points per term, maximum terms and anticipated credit points can exist only when intermission type flagged as studying at another institution.	Error
Approving person must be a staff member.	Error
Approving person cannot be the same as the student being approved.	Error

Table 2–13 lists the validations performed while creating student program award aim records. This information is useful for entering data into the IGS_EN_LGY_SPAA_INT table.

Table 2–13 Validations performed for creating student program award aim records

Validations	Validation Result (Warning/Error)
The same award aim must not exist for the same student program.	Warning
Start date must not be later than the end date.	Error
Award code must be offered within the enrolled program version.	Error

Table 2–14 lists the validations performed while creating thesis records. This information is useful for entering data into the IIGS_RE_LGCY_THE_INT table.

Table 2–14 Validations performed for creating thesis records

Validation	Validation Result (Error/Warning)
Matching thesis record must not already exist against the candidature. Thesis of same title is considered a duplicate, and the record is skipped.	Warning
Finalized indicator must be set if thesis has been submitted or examined.	Error
Embargo details cannot be recorded unless thesis has been submitted or examined.	Error
Embargo date cannot be specified without entering details of the embargo.	Error
Citation cannot be entered unless thesis has been submitted or examined.	Error
Library details cannot be entered unless thesis has been submitted or examined.	Error
Result cannot be entered until thesis has been examined.	Error
Result cannot be entered if current examination (submission) is outstanding.	Error
Result must be a final result.	Error
Expected submission date cannot be earlier than the minimum submission date.	Error
Expected submission date cannot be later than the maximum submission date.	Error
Submission date cannot be set if thesis title has not been finalized.	Error

Table 2–14 Validations performed for creating thesis records

Validation	Validation Result (Error/Warning)
Submission date cannot be set if no principal supervisor present (on the day of submission).	Error
Submission date cannot be set if no government type of activity code is set.	Error
Submission date cannot be earlier than program commencement date.	Error

Table 2–15 lists the validations performed while creating research supervisor records. This information is useful for entering data into the IGS_RE_LGCY_SPR_INT table.

Table 2–15 Validations performed for creating research supervisor records

Validation	Validation Result (Error/Warning)
Research supervisor must not already exist with the same start date. Duplicate record is one with the same supervising person with the same start date for the candidature, and it is skipped.	Warning
Research candidates cannot supervise themselves.	Error
If funding percentage is set, then supervising person must be a staff member.	Error
If funding percentage is set, then organizational unit must be set.	Error
If supervising person is a staff member, then funding percentage must be set.	Error
If supervising person is a staff member, then the organizational unit must be within the local institution.	Error
Supervision end date must be later than the start date.	Error
Supervision end date must be earlier than the start date of replacement supervisors.	Error
Records for the same supervisor must not overlap.	Error
A supervisor cannot replace themselves.	Error
A supervisor can only replace the latest instance of another supervisor.	Error
A replaced supervisor must have been ended.	Error
A replacement supervisor cannot overlap the replaced supervisor.	Error
The organization unit code must be in line with any specified organizational unit filter.	Error

Table 2–16 lists the validations performed while creating student program HESA statistics records. This information is useful for entering data into the IGS_HE_LGCY_SPA_INT table.

Table 2–16 Validations performed for creating student program HESA statistics records

Validation	Validation Result (Error/Warning)
Domicile code must exist as an OSS_DOM code.	Error
Occupation code must exist as an OSS_OCC code.	Error
Teacher training program ID must exist as an OSS_TTCID code.	Error
ITT phase must exist as an OSS_ITTPHSC code.	Error
Bilingual ITT Marker must exist as an OSS_BITTM code.	Error
Teaching qualification gain Sector must exist as an OSS_TQSEC code.	Error
Teaching qualification gain Subject 1 must exist as an OSS_TQSUB123 code.	Error
Teaching qualification gain Subject 2 must exist as an OSS_TQSUB123 code.	Error
Teaching qualification gain Subject 3 must exist as an OSS_TQSUB123 code.	Error
Destination must exist as an OSS_DEST code.	Error
ITT program outcome must exist as a OSS_OUTCOME code.	Error
NHS funding source must exist as an OSS_NHS_FUSRC code.	Error
UFI place must exist as an OSS_UFI_PLACE code.	Error
Social class indicator must exist as an OSS_SOC code.	Error
NHS employer must exist as a OSS_NHS_EMPLOYER code.	Error
Qualification aim proportion must exist as an OSS_PROPORTION code.	Error
Special student must exist as an OSS_SPEC_STUD code.	Error
FE student marker must exist as an OSS_FESTUMK code.	Error
Only a single HESA statistics detail must exist per program attempt.	Warning
Return type must exist as a IGS_HE_RED_RTN lookup.	Error
Student qualification aim must exist as an award available to the students program version.	Error
Student FE qualification aim must exist as an award available to the students program version.	Error

Table 2–16 Validations performed for creating student program HESA statistics records

Validation	Validation Result (Error/Warning)
Associated UCAS number can only be 'Y' or 'N'.	Error
Associated scottish candidate can only be 'Y' or 'N'.	Error
Associate teaching reference number can only be 'Y' or 'N'.	Error
Associate NHS registration number can only be 'Y' or 'N'.	Error
Subject qualification aim 1 must exist as a valid field of study.	Error
Subject qualification aim 2 can only be set if subject qualification aim 1 is also set.	Error
Subject qualification aim 2 must exist as a valid field of study.	Error
Subject qualification aim 3 can only be set if subject qualification aim 1 and subject qualification aim 2 are also set.	Error
Subject qualification aim 3 must exist as a valid field of study.	Error
Qualification aim proportion must be set if subject qualification 1 and 2 are set and subject qualification aim 3 is not set, otherwise it must be unset.	Error
Qualification aim proportion can only be set if subject qualification aim 1 and subject qualification aim 2 are also set.	Error
At any time either subject qualification aim 3 or qualification aim proportion can be set. You must not set both of them.	Error
If subject of qualification aim 1 is set and subject of qualification aim 2 is set, then either subject qualification aim 3 or qualification aim proportion must be set.	Error
The highest qualification on entry code must be a grade code for the grading schema defined with grading schema = value from IGS_HE_CODE_VALUES where code_type = HESA_HIGH_QUAL_ON_ENT.	Error

Table 2–17 lists the validations performed while creating student unit set HESA records. This information is useful for entering data into the IGS_HE_LGY_SUSA_INT table.

Table 2–17 Validations performed for creating student unit set HESA records

Validation	Validation Result (Error/Warning)
New HE entrant must exist as an OSS_HEENT code.	Error
Term time accommodation must exist as an OSS_TTA code.	Error
Disability allowance must exist as an OSS_DIS_ALLOW code.	Error
Additional support band must exist as an OSS_SUP_BAND code.	Error
SLDD discrete prov must exist as an OSS_ST13 code.	Error
Study mode must exist as an OSS_MODE_TYPE code.	Error
Study location must exist as an OSS_LOCSDY code.	Error
Franchising activity must exist as an OSS_FRAN_ACT code.	Error
Completion status must exist as an OSS_CSTAT code.	Error
Good standing marker must exist as an OSS_PROGRESS code.	Error
Complete PYR study code must exist as an OSS_FUNDCOMP code.	Error
Fundability code must exist as an OSS_FUND_CODE code.	Error
Fee eligibility must exist as an OSS_FEEELIG code.	Error
Fee band must exist as an OSS_FEEBAND code.	Error
Non payment reason must exist as an OSS_NONPAY code.	Error
Student fee must exist as an OSS_MSTUFEE code.	Error
Type of program year must exist as an OSS_TYPEYR code.	Error
Only a single HESA detail must exist per unit set attempt.	Warning
Can only be created against unit sets which are pre-enrollment year (i.e., YoP) sets.	Error
Credit level achieved 1 must exist as a unit level.	Error
Credit level achieved 2 must exist as a unit level.	Error
Credit level 1 must exist as a unit level.	Error
Credit level 2 must exist as a unit level.	Error

Table 2–17 Validations performed for creating student unit set HESA records

Validation	Validation Result (Error/Warning)
Grading schema grade must exist in the grading schema specified in the unit set statistics.	Error
Teaching institution 1 must exist as institution of type Post-Secondary.	Error
Teaching institution 2 must exist as institution of type Post-Secondary.	Error

Table 2–18 lists the validations performed while creating level advanced standing records for a program attempt. This information is useful for entering data into the IGS_AV_LGCY_LVL_INT table.

Table 2–18 Validations performed for creating level advanced standing records

Validation	Validation Result (Error/Warning)
The person must not be deceased.	Error
The exemption institution code entered must be an ACTIVE institution and must be present in the database.	Error
If the advanced standing is being approved for a local institution, then the total credit point granted must not exceed the program-version internal advanced-standing limit.	Error
If the advanced standing is being approved for an external institution, then the total credit point granted must not exceed the program-version external advanced-standing limit.	Error
Advanced standing unit level record must not already exist.	Error
Expiry date must be later than current system date.	Error
Approved date must not be earlier than the granted date, cancelled date or revoked date for the same record.	Error
Granted date must be earlier than the approved date.	Error
Authorizing person must be a valid staff person.	Error
Exception institution code must be valid.	Error
Advanced standing approved / granted must not exceed the advanced standing internal or external limits of the program version	Error
Person must not have a hold.	Error
Approved date must be earlier than expiry date.	Error

Table 2–18 Validations performed for creating level advanced standing records

Validation	Validation Result (Error/Warning)
Expiry date must be later than the approved date.	Error
Granted date must be earlier than the approved date.	Error
Cancelled date must be later than or the same as the approved date.	Error
Revoked date later than or the same as the approved date.	Error
One and only one of qualification details, unit details or test result details can be entered.	Error
The basis year must be outside the regency for the program version.	Error

Table 2–19 lists the validations performed while creating unit advanced standing records for a program attempt. This information is useful for entering data into the IGS_AV_LGCY_UNT_INT table.

Table 2–19 Validations performed for creating unit advanced standing records

Validation	Validation Result (Error/Warning)
Person must not be deceased.	Error
The exemption institution code entered must be an ACTIVE institution and must be present in the database.	Error
If the advanced standing is being approved for a local institution, then the total credit point granted must not exceed the program-version internal advanced-standing limit.	Error
If the advanced standing is being approved for an external institution, then the total credit point granted must not exceed the program-version external advanced-standing limit.	Error
Student unit advanced standing record must not already exist.	Error
If advanced standing recognition type is of type PRECLUSION, then CREDIT_PERCENTAGE cannot be specified.	Error
If advanced standing recognition is of type PRECLUSION, then credit point (ADV_STND_CREDIT_POINTS) must not be specified.	Error
Advanced standing of type of PRECLUSION cannot be granted.	Error
If advanced standing granting status is EXPIRED, then the expiry date must be specified.	Error
Total approved / granted credit point must not exceed the minimum credit points for program completion.	Error

Table 2–19 Validations performed for creating unit advanced standing records

Validation	Validation Result (Error/Warning)
Total approved / granted credit point must not exceed the program-version external advanced-standing limit.	Error
Total approved / granted credit point must not exceed the program-version internal advanced-standing limit.	Error
Person must not have a hold.	Error
A student cannot be granted advanced standing for a program in which the student has a program attempt but in a different version number.	Error
If the advanced standing status is GRANTED, then the GRANTED_DT must be specified.	Error
If the advanced standing status is CANCELLED, then the CANCELLED_DT must be specified.	Error
If the advanced standing status is REVOKED, then the revoked date must be specified.	Error
Approved date cannot be later than the expiration date.	Error
Granted date cannot be earlier than the approved date.	Error
The authorizing person must be a valid staff member.	Error
Either credit points or credit percentage must be specified.	Error
Credit points and credit percentage both must not be entered.	Error
If institution code is specified, then previous unit details must also be specified.	Error
Either institution code or test result details must be specified.	Error
Either previous unit details or test result details must be specified.	Error
If previous unit details are not specified then test result details cannot be specified.	Error
The grade must be a valid grade for the given grading schema.	Error
The term applied must be a valid term for a load calendar.	Error
One and only one of unit details or test result details must be entered.	Error
Alternate record must not already exist.	Warning
Alternate unit must not be the same as the unit code.	Error
Basis year must not be later than the current year.	Error

Table 2–20 lists the validations performed while creating unit outcomes records. This information is useful for entering data into the IGS_AS_LGCY_UOT_INT table.

Table 2–20 Validations performed for creating unit outcome records

Validations	Validation Result (Warning/Error)
Before assigning a grade to a student unit attempt the the system verifies if there is grading schema attached to the unit section level. If there is grading schema attached to the unit section and it is defaulted, then the student can be graded based on the same grading schema.	If there is no default grading schema attached in the unit section level, then the grading schema attached to the unit would be taken into consideration while assigning grades to the student unit attempt.
If the unit attempt status is not ENROLLED or COMPLETED, then the student unit attempt cannot be graded.	Error

Table 2–21 lists the validations performed while creating progression records. This information is useful for entering data into the IGS_AS_LGCY_PRO_INT table.

Table 2–21 Validations performed for creating progression records

Validation	Validation Result (Warning/Error)
Program attempt status cannot be Unconfirmed.	Error
Specified calendar must be an active progression calendar.	Error
The expiry date cannot be a future date.	Error
If decision status is Pending, then decision organizational unit or decision date cannot be set**.**	Error
If decision status is Approved or Waived, then decision organizational unit and decision date must be set.	Error
The decision date cannot be a future date.	Error
If progression outcome type is related to hold with SUS_PROGRAM (program suspension) or EXC_PROGRAM (program exclusion), then program must be specified.	Error
If progression outcome type is related to hold with EXC_PROG_US (exclude unit set), then unit set must be specified.	Error

Table 2–21 Validations performed for creating progression records

Validation	Validation Result (Warning/Error)
If progression outcome type is related to hold with EXC_PROG_U (exclude unit), then unit must be specified with unit type of EXCLUDED.	Error
If progression outcome type is related to hold with RQRD_PROG_U (required unit), then unit must be specified with unit type of REQUIRED.	Error
If progression outcome type is related to hold with RSTR_GE_CP or RSTR_LE_CP (restricted minimum or maximum CP threshold), then Restricted Enrollment CP must be set.	Error
If progression outcome type is related to hold with RSTR_AT_TY (restricted attendance type), then Restricted Attendance Type must be set.	Error
If progression outcome type is related to hold with EXC_PROG_GP (excluded program group), then Excluded Program Group must be set.	Error
The excluded program group can only be set when the progression outcome type is related to a hold with the EXC_PROG_GP (excluded program group) effect.	Error
The restricted enrollment CP can only be set when the progression outcome type is related to a hold with the RSTR_GE_CP or RSTR_LE_CP (restricted minimum or maximum CP) effect.	Error
The restricted attendance type can only be set when the progression outcome type is related to a hold with the RSTR_AT_TY (restricted attendance type) effect.	Error
If duration is set, then duration type must also be set and vice versa.	Error
If progression outcome type is a suspension, then duration must be set.	Error
If progression outcome type is an exclusion, suspension, or no penalty, then duration cannot be set.	Error
If progression outcome type is not probation or manual, then the duration type cannot be effective.	Error
If either the show cause outcome date or show cause outcome type are set, then the show cause date must also be set.	Error
The show cause outcome date cannot be a future date.	Error
If show cause outcome date is set, then show cause outcome type must also be set and vice versa	Error
The show cause date cannot be a future date.	Error
The show cause date cannot be set if the decision date is null (i.e., no decision is made).	Error
The show cause date cannot be set if the show cause expiry date is not set.	Error

Table 2–21 Validations performed for creating progression records

Validation	Validation Result (Warning/Error)
The show cause expiry date cannot be set if the decision date is null (i.e., no decision is made).	Error
The show cause expiry date cannot be later than the appeal expiry date.	Error
If either the appeal outcome date or appeal outcome type are set, then the appeal date must also be set.	Error
The appeal outcome date cannot be a future date.	Error
If the appeal outcome date is set, then the appeal outcome type must also be set and vice versa.	Error
The appeal date cannot be a future date.	Error
The appeal date cannot be earlier than the show cause date.	Error
The appeal expiry date cannot be set if the decision status is not Approved.	Error
Program can only be specified when the progression outcome type is related to hold with SUS_PROGRAM (program suspension) or EXC_PROGRAM (program exclusion).	Error
Unit set can only be specified when the progression outcome type is related to hold with EXC_PROG_US (unit set exclusion).	Error
Unit set version cannot be planned.	Error
Unit of type EXCLUDED can only be specified if the progression outcome type is related to hold with EXC_PROG_U (exclude unit).	Error
Unit of type REQUIRED can only be specified if the progression outcome type is related to hold with RQRD_PROG_U (required unit).	Error
Award must be a medal or a prize.	Error
Award can only be specified for positive progression outcomes.	Error

Table 2–22 lists the validations performed while creating class rank records. This information is useful for entering data into the IGS_PR_LGCY_CLR_INT table.

Table 2–22 Validations performed for creating class rank records

Validations	Validation Result (Warning/Error)
Student program attempt cannot be unconfirmed.	Error
The student must not already have a ranking within the class rank cohort instance.	Warning

Table 2–23 lists the validations performed while creating graduand records. This information is useful for entering data into the IGS_GR_LGCY_GRD_INT table.

Table 2–23 Validations performed for creating graduand records

Validation	Validation Result (Error/Warning)
Record with the same person ID and create date must not exist.	Warning
When the graduand status is GRADUATED or SURRENDER, then graduand type cannot be UNKNOWN or DEFFERED and conferral date must be set.	Error
If the conferral date is set, then graduand must be graduated.	Error
When a graduand is not linked to a student program attempt, then only an honorary award can be specified.	Error
When a graduand is not linked to a student program attempt, then honors level must not be specified for an HONORARY award.	Error
When graduand is linked to a student program attempt then program award must be specified.	Error
In case of an honorary award the surrendering for program award cannot be specified.	Error
The surrendering for program award can only be specified when surrendering or articulating an award. Otherwise the surrendering for program cannot be specified.	Error
If the proxy award person number is passed then the proxy award person number must be different to the graduand.	Error
If the proxy award person number is specified, then the proxy person cannot not be deceased.	Error
If the proxy award person number is passed, then the graduand type must be ATTENDING when proxy details are specified.	Warning

Table 2–23 Validations performed for creating graduand records

Validation	Validation Result (Error/Warning)
If proxy award person number is not specified and the graduand is deceased, then graduand requires a proxy to attend a ceremony.	Error
The graduand cannot be graduated unless the graduand approval status has a system value of APPROVED.	Error
If the student program attempt program requirements are not complete and it is not an approved alternative exit program award, graduand record cannot be created.	Error
The student program attempt program version must graduate students.	Error
The award must be of system award type Program for a program award.	Error
The award must be of system award type Honorary for an honorary award.	Error
System award type for a special award must be valid in case of special award.	Error
The program award must be an award for the student program attempt or an alternative exit.	Error
System graduand type must not be UNKNOWN or DEFERRED.	Error
If related graduand award ceremony records exist, then graduand type must be ATTENDING, INABSENTIA or UNKNOWN.	Error
Surrendering for program must not be the same as the surrendering program, if specified.	Error
Surrender for program must match a student program attempt belonging to the graduand.	Error
Conferral date must lie within the ceremony-round graduation calendar date range.	Error
If the graduation type Attending, then proxy award person number field value must be NULL.	Error

Enrollments and Records Legacy Data Interface Tables

Oracle Student System uses the following interface tables during the Import process.

- Batch details: IGS_EN_LGCY_BAT_INT
- Student Program Attempts: IGS_EN_LGCY_SPA_INT
- Student Unit Set Attempts: IGS_EN_LGY_SUSA_INT
- Student Program Intermission: IGS_EN_LGCY_SPI_INT
- Student Program Award Aim: IGS_EN_LGY_SPAA_INT
- Research Supervisors: IGS_RE_LGCY_SPR_INT
- Research Thesis: IGS_RE_LGCY_THE_INT
- Student Unit Attempt: IGS_EN_LGCY_SUA_INT
- Student Program HESA Statistics: IGS_HE_LGCY_SPA_INT
- Student Unit Set HESA Details: IGS_HE_LGY_SUSA_INT
- Advanced Standing Level: IGS_AV_LGCY_LVL_INT
- Unit Advanced Standing: IGS_AV_LGCY_UNT_INT
- Unit Outcomes: IGS_AS_LGCY_UOT_INT
- Progression Outcome: IGS_PR_LGCY_PRO _INT
- Class Rank: IGS_PR_LGCY_CLR_INT
- Graduand: IGS_GR_LGCY_GRD_INT
- Error messages: IGS_EN_LGCY_ERR_INT

For information on table column descriptions and values, see Oracle Student System Table Information at http://metalink.oracle.com/metalink/plsql/ml2_gui.startup.

Part II

Admissions Module

3

Admissions Import Process

This chapter describes the process of importing person and admission information into Oracle Student System. The following sections are in this chapter:

Definition

The Admissions Import process is a concurrent program that is used to collect and transfer the following types of person and admission information from interface tables to Oracle Student System target tables:

- person information
- admission application information
- test results
- transcripts

Overview

Admission applications and supporting documentation originate from various sources and in various formats. Application forms may be submitted in paper format, via the Internet, or as an electronic file from a third party. Supporting documentation provided by various sources, for example, institutions, testing agencies, government, professors, or alumni, may arrive before or after the application form is received.

The following is a partial list of admission application information that can be manually entered or imported:

- personal details
- academic history details
- relevant employment history details
- inquirer's or applicant's address
- characteristics of an inquiry
- collateral or brochures requested by inquirer
- program details
- unit or unit set inquiry or application details
- alternate names by which the applicant is known
- alternative codes used to identify applicant in other systems or organizations
- applicant's disability details
- statistical information for reporting purposes

Interface tables are provided by Oracle Student System to import the data. Institutions determine how to map source file data elements to the interface tables. The tables must be populated with data reviewed and standardized by the institution before importing it into Oracle Student System. Interface tables have columns that map to Oracle Student System data elements. Validation rules established in Oracle Student System limit the values that can be imported from the interface table into Oracle Student System.

The following topics are described in this section:

- Source Types
- Source Categories

- Match Set Criteria
- Data Element Discrepancies
- Duplicate Records
- Status Codes

Source Types

Source types are system-defined or user-defined classifications that describe the source of student data. System-defined source types include prospect lists, test results, admission applications, inquiries, and transcripts.

Admission application data files for import can originate from sources such as the College Board, College Net, EDI/SPEEDE, Educational Testing Service, Medical School Applications Board, a Centralized University Application Processing Center, or other third party application consolidators.

Some sources provide basic application information, such as name, address, and intended field of study. Other sources provide academic credential information, for example, test score results, academic transcript with grades earned, and courses taken or programs studied.

Users can set up various sources for importing data. Each of the user-defined sources are mapped to a system-defined source type as defined in the IGS_AD_SYSSRC_CAT table. Source categories are loaded from the seed data table to the IGS_AD_SOURCE_CAT_ALL table.

System-defined source types available for import in the Admissions subsystem are shown in Table 3–1, page 3-3.

Table 3–1 System-Defined Source Types Available for Import

Source Type	Description
PROSPECT_LIST	Prospective students; used to import person and person-related information
TEST_RESULTS	Student test results; used to import information related to test scores

Table 3–1 System-Defined Source Types Available for Import

Source Type	Description
APPLICATION	Student application details; used to import person and admission-related information
TRANSCRIPT	Student academic details; used to import transcript-related information
PROSPECT_SS_WEB_INQUIRY	Admission inquiries; used to import inquiries from prospective students

To review source types of application, transcripts, prospect list, test results, and prospect self service inquiry with their associated seeded source categories and corresponding interface tables, see Appendix B, "Admission Import Source Category Tables".

Source Categories

Source categories are logical groups of attributes used to define rules when importing prospect or admission records from an outside source.

Data received from various sources needs to be imported into Oracle Student System. Interface tables are provided for importing data. Institutions determine how to map source file data elements to the interface tables in the Source Categories window. The interface tables are populated with data that the institution must review and standardize before importing into Oracle Student System.

For example, if an institution has standardized country codes such as GB for the United Kingdom in Oracle Applications, the institution must validate that the data contains country code GB only and not UK.

All system-defined source types have a valid list of source categories, some of which are mandatory, while others are optional. When user-defined source types are set up, the mandatory categories for the corresponding system-defined source types are automatically included from the table IGS_AD_SYSCAT_ATTR.

Match Set Criteria

Match set criteria are used to identify exact and partial record matches while importing person data. Before importing external data, Oracle Student System checks if the import data already exists in the system. If the data to be imported already exists in the system, depending on the discrepancy rules set for the source category, the data may need to be reviewed or may not be imported. The user needs to define the criteria used to match information to be imported against information that already exists in the system.

During the Admissions Import process, a matching process takes place to identify matched person records. If a matched person record is identified, data in the import file is used to update the record in accordance with defined discrepancy rules. If a matched person record is not found, a new record is created. Records that do not match all of the required data element values are known as near or partial matches. Partial matches are listed in the Exact/Partial Match Report and the Partial Matching Records window. Partially matched records are not updated and the Admissions Import process moves on to the next record. Users can match records for existing persons in the system using the Partial Matching Records. Once any matches have been identified, the interface tables can be updated and the Admissions Import process can be run again.

For information on partial matching records, see Chapter 4, Admissions, *Oracle Student System User's Guide.*

A default set of elements used for identifying exact matches is provided in Oracle Student System. A default set of elements is also provided for identifying partial matches. Users can customize the default sets and create new sets. A one-to-one relationship must always exist between an exact match criteria set or a partial match criteria set.

The complete set of data elements that can be used when creating match criteria sets is as follows:

- surname
- given name
- preferred alternate ID
- alternate ID
- person type
- date of birth
- sex
- ethnic origin
- address line 1
- address line 2
- address line 3

- address line 4
- address type
- city
- state
- province
- postcode
- county
- country
- given name, first character
- surname, first five characters

Table 3–2, page 3-6 shows the default match criteria set for exact or partial matches.

Table 3–2 Default Match Criteria Sets

Exact Match Set	Partial or Near Match Set
Surname	Surname
Given name, first character	Given name, first character

Users can create new custom match criteria by updating the exact, drop if null, and partial indicators for the existing data elements. Any of the data elements that can be used when creating match criteria sets can be added.

Table 3–3, page 3-6, describes the various matches that can occur and the match indicator codes associated with them.

Table 3–3 Match Indicators

Codes	Description
11	No match
12	Match to a single person
13	Match to multiple persons
14	Partial match; to be reviewed
15	Partial match; reviewed and matched to a single person ID. Needs to be imported.

Table 3–3 Match Indicators

Codes	Description
16	Partial match; reviewed and not matched to a person. Needs to be imported as a new person.
17	Partial match; reviewed. Do not import.
18	Match occurred and import values used.
19	Match exists and existing values retained.
20	Requires review and is pending.
21	Match reviewed; to be imported.
22	Match reviewed. Do not import.
23	Match reviewed; no discrepancy and existing values retained.
24	Invalid record

For information on the Exact/Partial Match Report, see Importing Admissions Information, Chapter 4, Admissions, *Oracle Student System User's Guide*.

Data Element Discrepancies

Each prospect or applicant must have only one record in Oracle Student System at any time. Different values may be received for the same data element in the applicant's record as a result of importing data from many sources. If data in the import file can be matched to an existing person record and new unique data element values exist, the person record is updated. If data in the import file can be matched to an existing person record and data element values conflict and the discrepancy rules are set to review details, an error code will be returned and an element discrepancy review report must be generated for the user to review.

Users can review the Import Process Review Report detailing the existing value and imported value for the record. The user can then decide which value to accept.

Table 3–4, page 3-8 shows various record matches and the action taken by Oracle Student System as a result of the match.

Table 3–4 Record Matches

Type of Record Match	Action Required
Record match identified with new data element values.	Update existing record.
Record match identified with conflicting data element values.	Generate Import Process Review report. Keep existing values or overwrite existing values, depending on discrepancy rules.
Record match identified with identical data element values.	No action required.
Record match not identified.	Create record.
Record near match identified.	Generate Exact / Partial Match Report or use partial matching records.

Duplicate Records

Duplicate records may be created when application data and supporting documentation are imported. Duplicate records can be identified by running the Identify Duplicate Persons process and the Review Duplicate Person Records Report. The user can choose to merge duplicate records if required.

For information on identifying and merging duplicate records, see Chapter 3, Person, *Oracle Student System User's Guide.*

Status Codes

After the Admissions Import process is complete, all records are assigned a status. The status code indicates if the record was imported successfully, has an error, had problems during the Admissions Import process, or is pending.

Table 3–5, page 3-9 lists the record status codes.

Table 3–5 Status Codes

Status Code	Description
1	Completed
2	Pending
3	Error
4	Warning

Prerequisites

The following setup steps must be completed before person and admission information can be imported into the Admissions module of Oracle Student System.

- ❏ Source types must be defined.
- ❏ Source categories must be defined.
- ❏ Match set criteria must be defined.
- ❏ Person Reference setup steps must be completed.
- ❏ Admissions setup steps must be completed.
- ❏ Calendar setup steps must be completed.
- ❏ Organizational structure setup steps must be completed.
- ❏ Program structure and planning setup steps must be completed.

Admissions Import Process Overview Diagram

Figure 3–1, page 3-10 shows the process of importing external admission data into Oracle Student System as described in the Admissions Import Process Overview section.

Figure 3–1 Admissions Import Process Overview Diagram

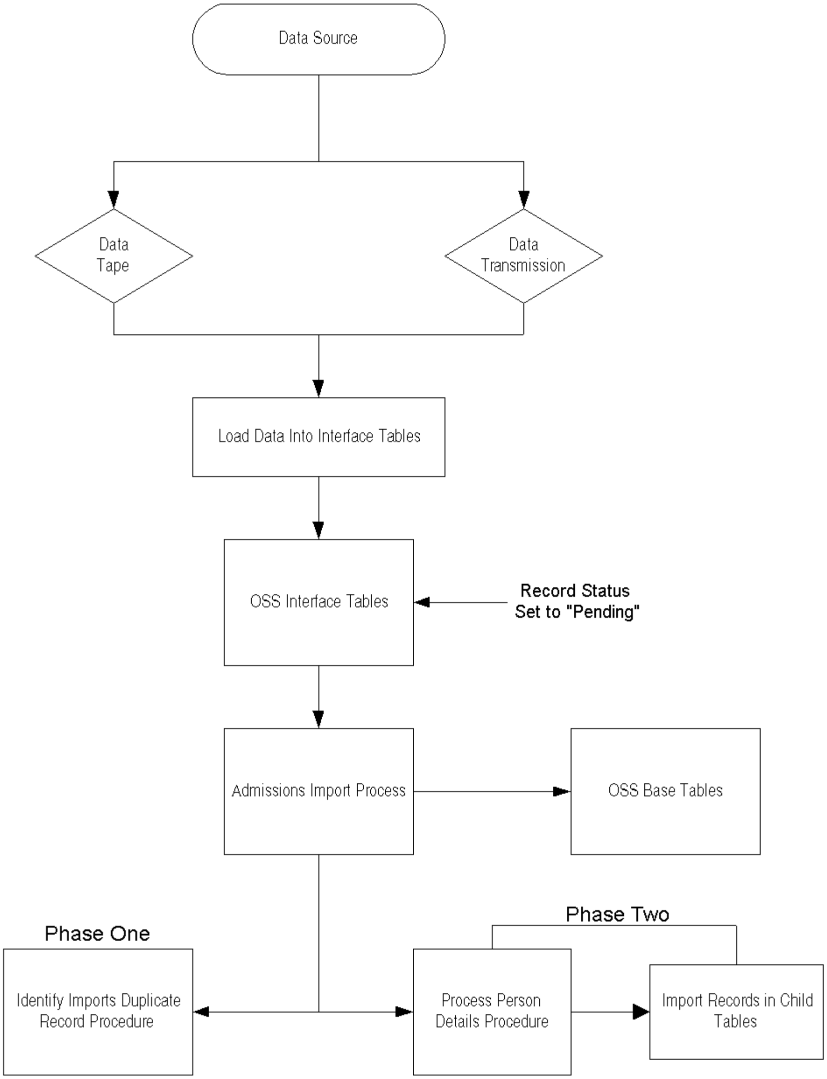

Data Source
Data Tape
Data Transmission
Load Data Into Interface Tables
OSS Interface Tables
Record Status Set to "Pending"
Admissions Import Process
OSS Base Tables
Phase One
Identify Imports Duplicate Record Procedure
Phase Two
Process Person Details Procedure
Import Records in Child Tables

Figure 3–2, page 3-11 and Figure 3–3 , page 3-12 shows the details of phase one and phase two of the admissions import process as described in the Admissions Import Process Overview section.

Figure 3–2 Phase One of the Admissions Import Process

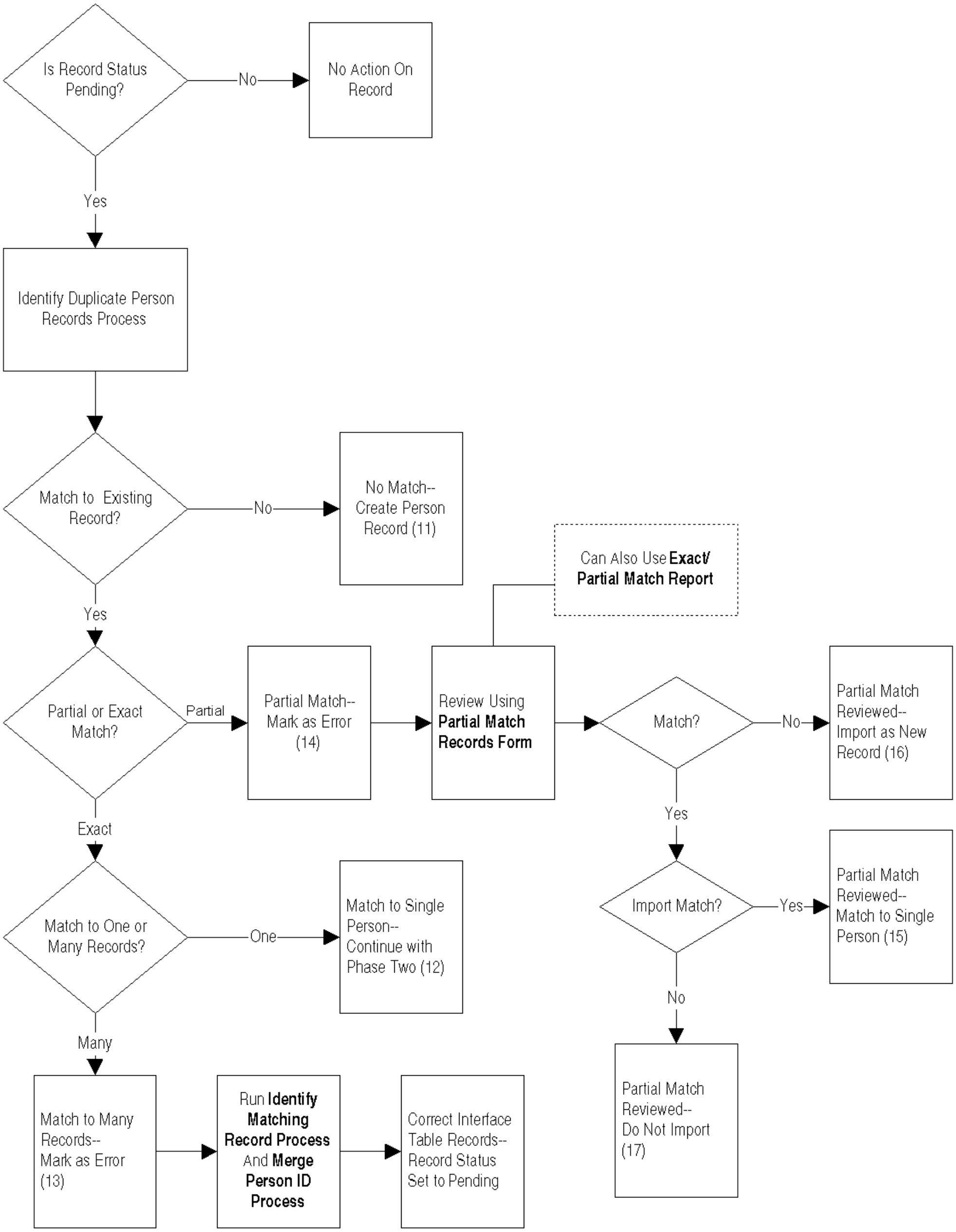
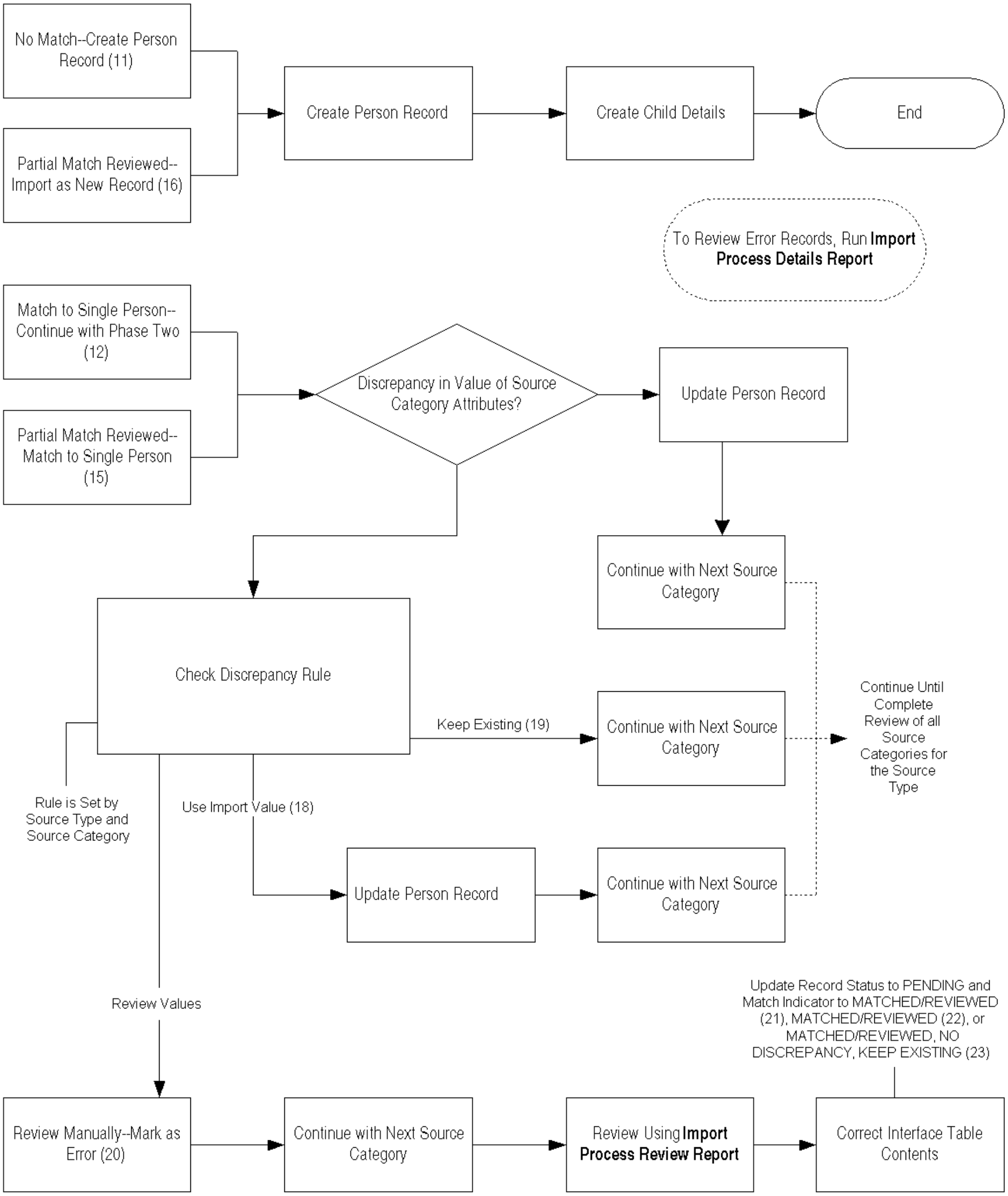

Is Record Status Pending?
No
No Action On Record
Yes
Identify Duplicate Person Records Process
Match to Existing Record?
No
No Match-- Create Person Record (11)
Can Also Use Exact/ Partial Match Report
Yes
Partial or Exact Match?
Partial
Partial Match-- Mark as Error (14)
Review Using Partial Match Records Form
Match?
No
Partial Match Reviewed-- Import as New Record (16)
Yes
Import Match?
Yes
Partial Match Reviewed-- Match to Single Person (15)
No
Partial Match Reviewed-- Do Not Import (17)
Exact
Match to One or Many Records?
One
Match to Single Person-- Continue with Phase Two (12)
Many
Match to Many Records-- Mark as Error (13)
Run Identify Matching Record Process And Merge Person ID Process
Correct Interface Table Records-- Record Status Set to Pending

Figure 3–3 Phase Two of the Admissions Import Process


No Match--Create Person Record (11)
Partial Match Reviewed--Import as New Record (16)
Create Person Record
Create Child Details
End
To Review Error Records, Run **Import Process Details Report**
Match to Single Person--Continue with Phase Two (12)
Partial Match Reviewed--Match to Single Person (15)
Discrepancy in Value of Source Category Attributes?
Update Person Record
Continue with Next Source Category
Check Discrepancy Rule
Keep Existing (19)
Continue with Next Source Category
Continue Until Complete Review of all Source Categories for the Source Type
Rule is Set by Source Type and Source Category
Use Import Value (18)
Update Person Record
Continue with Next Source Category
Review Values
Update Record Status to PENDING and Match Indicator to MATCHED/REVIEWED (21), MATCHED/REVIEWED (22), or MATCHED/REVIEWED, NO DISCREPANCY, KEEP EXISTING (23)
Review Manually--Mark as Error (20)
Continue with Next Source Category
Review Using **Import Process Review Report**
Correct Interface Table Contents

Admissions Import Process Overview

The Admissions Import process consists of the following procedures:

- Setting Up the Admissions Import Process
- Identify Imports Duplicate Record Procedure - Phase One
- Process Person Details Procedure - Phase Two
- Importing Records in Child Tables
- Generating Error Reports
- Correcting Errors

Setting Up the Admissions Import Process

Before the Admissions Import process begins, records in all interface tables are populated and set with an initial status of 2, Pending. The record status in the master interface table, IGS_AD_INTERFACE_ALL, is also set to Pending.

Note: The status of the IGS_AD_INTERFACE_ALL table, as well as the status of the record, is set to 1 (completed) by the import process. If there is an error in the import of any child table, the record status in the IGS_AD_INTERFACE_ALL table must be changed to 2 (pending) before running the import process again.

All records in IGS_AD_INTERFACE_ALL must have an associated source_type_id and batch_id. For a single Admissions Import process, the records to be imported must have the same BATCH_ID and SOURCE_TYPE_ID values. The sequence IGS_AD_INTERFACE_BATCH_ID_S is used to generate a unique sequence number for the batch_id column. All records that are populated in the interface table must have a unique interface_id. This id can be generated using the sequence IGS_AD_INTERFACE_ID_S.

Identify Imports Duplicate Record Procedure - Phase One

When the concurrent program is initiated, the Identify Imports Duplicate Record procedure is executed. This is the first pass of the Admissions Import process. The Admissions Import process identifies all records in IGS_AD_INTERFACE_ALL with a status of Pending and checks each record for a match according to the specified match criteria. Each record is updated with one of the following match indicators:

- 11: No match; person to be created
- 12: Match to single person
- 13: Match to duplicate persons; duplicate record
- 14: Partial match; to be reviewed

Master interface records with a match indicator of 13 or 14 are marked as errors and the child interface records for these records are not processed.

Process Person Details Procedure - Phase Two

When the Identify Imports Duplicate Record procedure is complete, the Process Person Details procedure is executed. This is the second pass of the Admissions Import process. The procedure identifies records with one of the following match indicators:

- 11: No match; person to be created
- 12: Match to single person
- 15: Partial match; reviewed and matched to a single person, to be imported
- 16: Partial match; reviewed and imported as new record

Within each record, a person record is either created or updated. If the record is updated, the discrepancy rule specified for the source type and source category PERSON details is checked, and the update is completed. The record can be marked for review if required. The action specified in the PERSON_MATCH_ID column states whether the record is created, updated, or reviewed. Records can also be marked as partially matched.

Records marked for review can be viewed by running the Import Review report. If a reviewed record is discarded, the status is changed to 2, Pending, and the match indicator is changed to 22, Matched, Reviewed, Discard. If a reviewed record is marked for import, the status is changed to 2 and the match indicator is changed to 21, Matched, Reviewed, Import. The Admissions Import process must be run again to process the corrections.

Any partially matched records can be viewed in the Partial Matching Records window. These records can be imported, discarded, or created as a new person record. The match indicator is set to one of the following:

- 15: Partial match; reviewed and matched to a single person, to be imported
- 16: Partial match; reviewed and imported as new record
- 17: Partial match; reviewed, do not import

Records marked as 15 and 16 are imported when the concurrent manager is next run.

If any errors are generated during the creation or update of a person record, the master interface record is marked with a relevant error code. No child interface records of this record are processed.

Creating an Admission Inquiry

At the end of the admission import process, depending on the source type parameter specified by the user, the Create Inquiry Instance subprocess is called, which creates an admission inquiry record for the relevant persons.

Importing Records in Child Tables

When a master record is successfully imported from IGS_AD_INTERFACE_ALL, the status column in this table for this row is set to 1. The processes which import information such as address, test results, and application details, only import for records where the master record status is 1.

Records in child tables of IGS_AD_INTERFACE_ALL have the following columns:

- MATCH_IND
- STATUS
- ERROR_CODE

MATCH_IND specifies whether the child record exists in the system and requires updating, or whether the record must be created. STATUS indicates the status of the creation or update process. If this process fails, the appropriate error code is updated in the ERROR_CODE column. If the failure is due to incorrect data or system fatal errors, the record is ignored and the next record is processed.

If a record is updated, the discrepancy rule specified for the source type and source category is checked and the update takes place accordingly. The record can be marked for review if required; the action taken is specified in the MATCH_IND column as create, update, or review.

Records marked for review are listed when the Import Review report is run. If a reviewed record is discarded, the status is changed to 2, Pending, and the match indicator is changed to 22, Matched, Reviewed, Discard. If a reviewed record is marked for import, the status is changed to 2 and the match indicator is changed to 21, Matched, Reviewed, Import. The Admissions Import process must be run again to process the corrections.

Generating Error Reports

Table 3–6, page 3-16, lists error reports by admission data source type, which can be generated to list error details.

Table 3–6 Error Reports Available for Admission Data Source Types

Admission Data Source Type	Error Report
Application	Import Process Person Details Report
	Import Process Application Details Report
Prospect List	Import Process Person Details Report
Test Results	Import Process Person Details Report
	Import Process Test Results Review Report
	Import Process Test Results Details Report
Transcripts	Import Process Person Details Report
	Import Process Transcript Details Review Report
	Transcript Details Error Report
Prospect Self Service Web Inquiry	Import Process Person Details Report
	Import Process Inquiry Details Report

For information on the Import Process Person Details Report, see Chapter 4, Admissions, *Oracle Student System User's Guide*.

For information on the Import Process Application Details Report, see Chapter 4, Admissions, *Oracle Student System User's Guide*.

For information on the Import Process Test Results Review Report, see Chapter 4, Admissions, *Oracle Student System User's Guide*.

For information on the Import Process Test Results Details Report, see Chapter 4, Admissions, *Oracle Student System User's Guide*.

For information on the Import Process Transcript Details Review Report, see Chapter 4, Admissions, *Oracle Student System User's Guide*.

For information on the Transcript Details Error Report, see Chapter 4, Admissions, *Oracle Student System User's Guide*.

For information on the Import Process Inquiry Details Report, see Chapter 4, Admissions, *Oracle Student System User's Guide*.

Correcting Errors

Error records have a status of 3. Interface tables errors must be corrected manually. Once errors are corrected, the record status must be changed to 2 = Pending. The Admissions Import process must be re-run to process corrections.

Note: Specify F as the value in the Sex field instead of Female.

Master and Child Admission Module Tables

Figure 3–4, page 3-18, Figure 3–5, page 3-19, and Table 3–6, page 3-20 show the master table, IGS_AD_INTERFACE_ALL and the various child tables associated with IGS_AD_INTERFACE_ALL which hold the person details to be imported.

Figure 3–4 Interface Table Structure

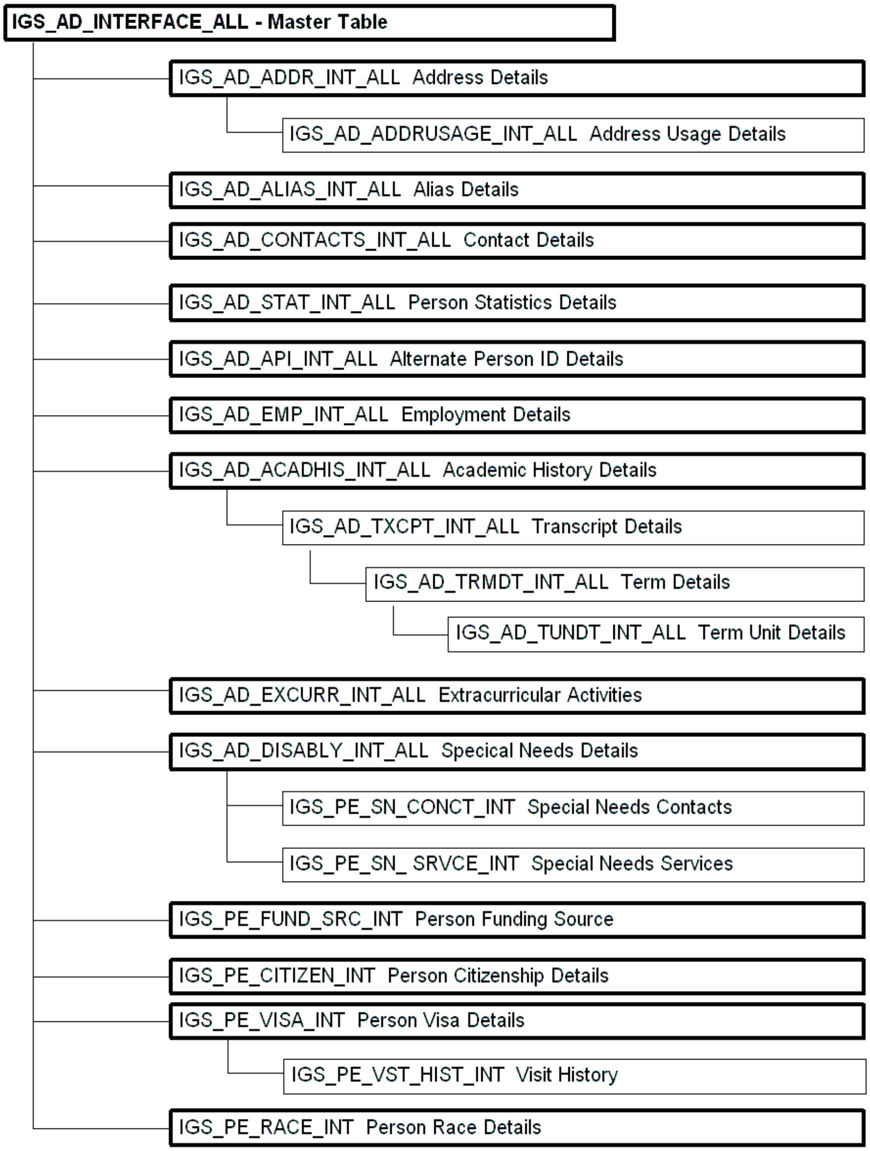

IGS_AD_INTERFACE_ALL - Master Table
IGS_AD_ADDR_INT_ALL Address Details
IGS_AD_ADDRUSAGE_INT_ALL Address Usage Details
IGS_AD_ALIAS_INT_ALL Alias Details
IGS_AD_CONTACTS_INT_ALL Contact Details
IGS_AD_STAT_INT_ALL Person Statistics Details
IGS_AD_API_INT_ALL Alternate Person ID Details
IGS_AD_EMP_INT_ALL Employment Details
IGS_AD_ACADHIS_INT_ALL Academic History Details
IGS_AD_TXCPT_INT_ALL Transcript Details
IGS_AD_TRMDT_INT_ALL Term Details
IGS_AD_TUNDT_INT_ALL Term Unit Details
IGS_AD_EXCURR_INT_ALL Extracurricular Activities
IGS_AD_DISABLY_INT_ALL Specical Needs Details
IGS_PE_SN_CONCT_INT Special Needs Contacts
IGS_PE_SN_ SRVCE_INT Special Needs Services
IGS_PE_FUND_SRC_INT Person Funding Source
IGS_PE_CITIZEN_INT Person Citizenship Details
IGS_PE_VISA_INT Person Visa Details
IGS_PE_VST_HIST_INT Visit History
IGS_PE_RACE_INT Person Race Details

Figure 3–5 Interface Table Structure Continued

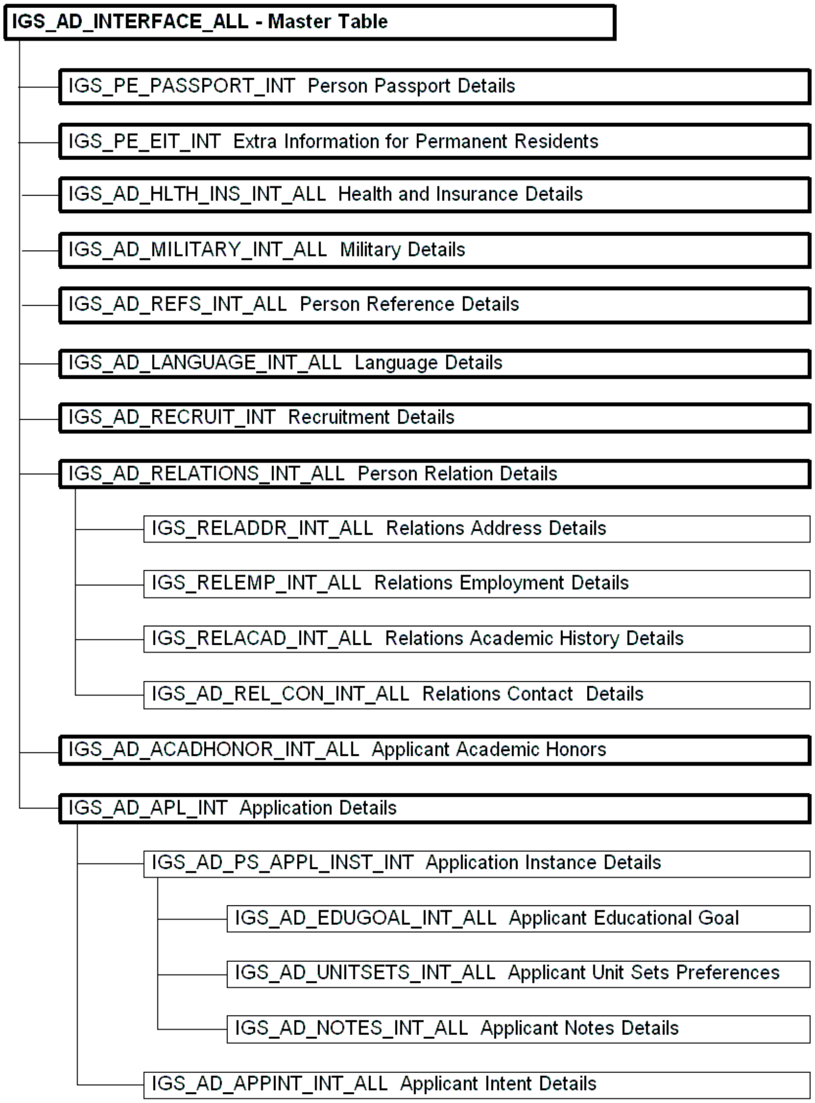
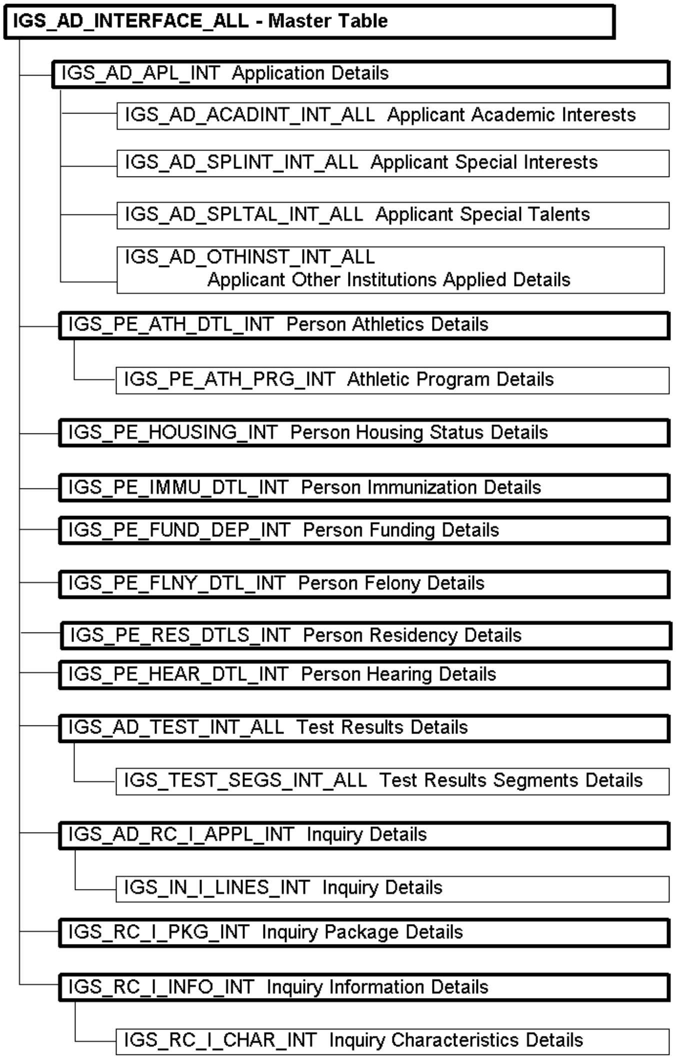

IGS_AD_INTERFACE_ALL - Master Table
IGS_PE_PASSPORT_INT Person Passport Details
IGS_PE_EIT_INT Extra Information for Permanent Residents
IGS_AD_HLTH_INS_INT_ALL Health and Insurance Details
IGS_AD_MILITARY_INT_ALL Military Details
IGS_AD_REFS_INT_ALL Person Reference Details
IGS_AD_LANGUAGE_INT_ALL Language Details
IGS_AD_RECRUIT_INT Recruitment Details
IGS_AD_RELATIONS_INT_ALL Person Relation Details
IGS_RELADDR_INT_ALL Relations Address Details
IGS_RELEMP_INT_ALL Relations Employment Details
IGS_RELACAD_INT_ALL Relations Academic History Details
IGS_AD_REL_CON_INT_ALL Relations Contact Details
IGS_AD_ACADHONOR_INT_ALL Applicant Academic Honors
IGS_AD_APL_INT Application Details
IGS_AD_PS_APPL_INST_INT Application Instance Details
IGS_AD_EDUGOAL_INT_ALL Applicant Educational Goal
IGS_AD_UNITSETS_INT_ALL Applicant Unit Sets Preferences
IGS_AD_NOTES_INT_ALL Applicant Notes Details
IGS_AD_APPINT_INT_ALL Applicant Intent Details

Figure 3–6 Interface Table Structure Continued


IGS_AD_INTERFACE_ALL - Master Table
IGS_AD_APL_INT Application Details
IGS_AD_ACADINT_INT_ALL Applicant Academic Interests
IGS_AD_SPLINT_INT_ALL Applicant Special Interests
IGS_AD_SPLTAL_INT_ALL Applicant Special Talents
IGS_AD_OTHINST_INT_ALL Applicant Other Institutions Applied Details
IGS_PE_ATH_DTL_INT Person Athletics Details
IGS_PE_ATH_PRG_INT Athletic Program Details
IGS_PE_HOUSING_INT Person Housing Status Details
IGS_PE_IMMU_DTL_INT Person Immunization Details
IGS_PE_FUND_DEP_INT Person Funding Details
IGS_PE_FLNY_DTL_INT Person Felony Details
IGS_PE_RES_DTLS_INT Person Residency Details
IGS_PE_HEAR_DTL_INT Person Hearing Details
IGS_AD_TEST_INT_ALL Test Results Details
IGS_TEST_SEGS_INT_ALL Test Results Segments Details
IGS_AD_RC_I_APPL_INT Inquiry Details
IGS_IN_I_LINES_INT Inquiry Details
IGS_RC_I_PKG_INT Inquiry Package Details
IGS_RC_I_INFO_INT Inquiry Information Details
IGS_RC_I_CHAR_INT Inquiry Characteristics Details

Oracle Student System Interface Tables

Oracle Student System uses the following interface tables during the Admissions Import process:

- IGS_AD_INTERFACE_ALL
- IGS_AD_STAT_INT_ALL
- IGS_PE_EIT_INT (when Information Type=PE_INT_PERM_RES)
- IGS_AD_ACADHIS_INT_ALL
- IGS_AD_ADDR_INT_ALL
- IGS_AD_ADDRUSAGE_INT_ALL
- IGS_AD_ALIAS_INT_ALL
- IGS_PE_PASSPORT_INT
- IGS_PE_RES_DTLS_INT
- IGS_PE_ATH_DTL_INT
- IGS_PE_ATH_PRG_INT
- IGS_AD_CONTACT_INT_ALL
- IGS_PE_RACE_INT
- IGS_PE_VISA_INT
- IGS_PE_FLNY_DTL_INT
- IGS_PE_HEAR_DTL_INT
- IGS_AD_EMP_INT_ALL
- IGS_AD_EXCURR_INT_ALL
- IGS_AD_HLTH_INS_INT_ALL
- IGS_PE_IMMU_DTL_INT
- IGS_PE_HOUSING_INT
- IGS_AD_API_INT_ALL
- IGS_PE_FUND_SRC_INT
- IGS_PE_CITIZEN_INT
- IGS_AD_LANGUAGE_INT_ALL

- IGS_AD_MILITARY_INT_ALL
- IGS_AD_RECRUIT_INT
- IGS_AD_DISABLTY_INT_ALL
- IGS_PE_SN_CONCT_INT
- IGS_PE_SN_SRVCE_INT
- IGS_AD_RELATIONS_INT_ALL
- IGS_AD_RELACAD_INT_ALL
- IGS_AD_RELADDR_INT_ALL
- IGS_AD_REL_CON_INT_ALL
- IGS_AD_RELEMP_INT_ALL
- IGS_AD_ACADINT_INT_ALL
- IGS_AD_ACADHONOR_INT_ALL
- IGS_AD_EDUGOAL_INT_ALL
- IGS_AD_APPINT_INT_ALL
- IGS_AD_NOTES_INT_ALL
- IGS_AD_OTHINST_INT_ALL
- IGS_AD_SPLINT_INT_ALL
- IGS_AD_SPLTAL_INT_ALL
- IGS_AD_UNITSETS_INT_ALL
- IGS_AD_APL_INT
- IGS_AD_PS_APPL_INST_INT
- IGS_RC_I_APPL_INT
- IGS_RC_I_CHAR_INT
- IGS_RC_I_INFO_INT
- IGS_RC_I_PKG_INT
- IGS_AD_TXCPT_INT_ALL
- IGS_AD_TRMDT_INT_ALL
- IGS_AD_TUNDT_INT_ALL

For information on table column descriptions and values, see Oracle Student System Table Information at http://metalink.oracle.com/metalink/plsql/ml2_gui.startup.

Admission Import Process Validations

The following tables list the validations performed on various data elements imported into Oracle Student System during the admission import process.

Table 3–7 Validations performed on data imported from the table IGS_AD_INTERFACE_ALL

Column Name	Validation performed
PRE_NAME_ADJUNCT	Must be an active record in FND_LOOKUP_VALUES, for lookup type='CONTACT_TITLE'
SEX	Must be an active record in IGS_LOOKUPS_VIEW, for lookup type='SEX'
BIRTH_DT	If Not null, then must be earlier than current system date.
PERSON_NUMBER	If HZ_GENERATE_PARTY_NUMBER profile = N then Person Number is mandatory.
PREF_ALTERNATE_ID	Must match the value of preferred_ind in IGS_PE_PERSON_ID_TYP.
PROOF_OF_INS	Format Mask of the value for this field, must be same as specified in IGS_PE_PERSON_ID_TYP.
	Must be Y or N.
PROOF_OF_IMMU	Must be Y or N.
MILITARY_SERVICE_REG	Must be Y or N.
BIRTH_COUNTRY	Must be an active record in FND_LOOKUP_VALUES_VL, for lookup type= PER_US_COUNTRY_CODE.
LEVEL_OF_QUAL_ID	Must be an active record in IGS_AD_CODE_CLASSES, for class ='’LEVEL_OF_QUAL'.
VETERAN	Must be an active record in IGS_LOOKUPS_VIEW, for lookup type=' VETERAN_STATUS'.

Table 3–8 Validations performed on data imported from the table IGS_AD_STAT_INT

Column Name	Validation performed
RELIGION_CD	SHould be an active record in IGS_LOOKUP_VALUES, for lookup type= PE_RELIGION.
Flexfield Columns, if provided	If a value is passed, then the descriptive flexfield, IGS_PE_PERS_STAT must have been configured.
MARITAL_STATUS	Must be an active record in FND_LOOKUP_VALUES for lookup 'MARITAL_STATUS'.
MARITAL_STATUS_EFFECTIVE_DATE	If not null, must be later than DOB of the person.
ETHNIC_ORIGIN	Must be an active record in IGS_LOOKUPS_VIEW for lookup type = IGS_ETHNIC_ORIGIN.

Table 3–9 Validations performed on data imported from the table IGS_AD_API_INT_ALL

Column Name	Validation performed
Flexfield columns, if provided	If a value is passed, then the descriptive flexfield, IGS_PE_ALT_PERS_ID_FLEX must be configured.
PERSON_ID_TYPE	Must match a record in IGS_PE_PERSON_ID_TYP for PERSON_ID_TYPE.
	Format Mask must be same as set in IGS_PE_PERSON_ID_TYP.
REGION_IND	If REGION_IND is null or ‘N’ and REGION_CD in IGS_AD_API_INT_ALL must not be null
	IF REGION_IND is null or ‘N’ then REGION_CD must not be provided
REGION_CD	If REGION_IND is not null, then REGION_CD must not be null.
	REGION_CD must be a STATE in AR_LOC_ONE_LEVEL_V

Table 3–10 Validations performed on data imported from the table IGS_AD_ADDR_INT_ALL

Column Name	Validation performed
START_DATE	Start date must not be null. If birth date is not null for the IGS_AD_INTERFACE_ALL record, then it must be earlier than the start date.

Table 3–10 Validations performed on data imported from the table IGS_AD_ADDR_INT_ALL

Column Name	Validation performed
END_DATE	END date must be later than the start date.
COUNTRY	Validated from territory short name in FND_TERRITORIES_VL
CORRESPONDENCE_FLAG	Must be in Y, N

Table 3–11 Validation performed on data imported from the table IGS_AD_ADDRUSAGE_INT_ALL

Column Name	Validation performed
SITE_USE_CODE	Must match a record in FND_LOOKUP_VALUES for lookup type = PARTY_SITE_USE_CODE.

Table 3–12 Validations performed on data imported from the table IGS_PE_ATH_DTL_INT

Column Name	Validation performed
ELIGIBILITY_STATUS_CD	If not null, must match an active record in IGS_LOOKUP_VALUES for lookup type = PE_ATH_ELG_STATUS.
TENTATIVE_ADM_CODE	If not null, an active record must be present in IGS_LOOKUP_VALUES for lookup type = PE_ATH_ELG_STATUS and lookup code = eligibility_status_cd.
PREDICT_ELIG_CODE	If not null, an active record must be present in IGS_LOOKUP_VALUES for lookup type = PE_ATH_ELG_STATUS and lookup code = ELIGIBILITY_STATUS_CD.
ATHLETIC_GPA	If not null, must be greater than 0.
REVIEW_DATE	If review date and date of birth are both not null, then the review date must be later than the date of birth.

Table 3–13 Validations performed on data imported from the table IGS_PE_ATH_PRG_INT

Column Name	Validation performed
ATHLETIC_PRG_CODE	Must not be null, and must match an active record in IGS_LOOKUP_VALUES for lookup type = PE_ATH_PRG_TYPE.
START_DATE	Start date must be later than the date of birth.
START_DATE, END_DATE	Start date must be earlier than the end date.
RECRUITED_IND	Must be either ‘Y’ or ‘N’.
PARTICIPATING_IND	Must be either ‘Y’ or ‘N’.

Table 3–14 Validations performed on data imported from the table IGS_AD_ACADHIS_INT_ALL

Column Name	Validation performed
INSTITUTION_CODE	If not null, this must match a record in IGS_OR_INSTITUTION for active institutions.
PROGRAM_TYPE_ATTEMPTED	If not null, this must match a record in IGS_PS_TYPE.
PROGRAM_TYPE_EARNED	If not null, this must match a record in IGS_PS_TYPE.
PROGRAM_CODE	If not null, must match a record in IGS_PS_VER.
START_DATE	Start date must be later than the system date.
START_DATE, END_DATE	If both are not null, then the start date must be earlier than the end date.
START_DATE, END_DATE	If end date is not null, then the start date must not be null.
PLANNED_COMPLETION_DATE, START_DATE	If the planned completion date is not null, it must be later than the start date.
SELFREP_TOTAL_CP_ATTEMPTED	If not null, this must be greater than or equal to 0.
SELFREP_TOTAL_CP_EARNED	If not null, this must be greater than or equal to 0.
SELFREP_TOTAL_GP_UNITS_ATTEMP	If not null, this must be greater than or equal to 0.
SELFREP_INST_GPA	If not null, this must be greater than or equal to 0.
SELFREP_GRADING_SCALE_ID	If not null, this must be greater than 0.
SELFREP_WEIGHTED_GPA	If not null, this must be either ‘Y’ or ‘N’.
SELFREP_RANK_IN_CLASS	If not null, this must be greater than 0.
SELFREP_WEIGHTED_RANK	If not null, this must be either ‘Y’ or ‘N’.

Table 3–14 Validations performed on data imported from the table IGS_AD_ACADHIS_INT_ALL

Column Name	Validation performed
CLASS_SIZE	If not null, then must be greater than 0.
Flexfield Columns, if provided	If value passed, then the descriptive flexfield, IGS_AD_ACAD_HISTORY_FLEX must have been configured.
TRANSCRIPT_REQUIRED	If not null, this must be either ‘Y’ or ‘N’.

Table 3–15 Validations performed on data imported from the table IGS_AD_ALIAS_INT

Column Name	Validation performed
ALIAS_TYPE	Must be an active record in lookup type PE_ALIAS_TYPE in IGS_LOOKUP_VALUES.
START_DT	Start date must be later than the birth date.
END_DT	End date must be later than the start date.

Table 3–16 Validations performed on data imported from the table IGS_AD_EMP_INT_ALL

Column Name	Validation performed
OCCUPATIONAL_TITLE_CODE	Validate occupation title from IGS_PS_DIC_OCC_TITLS.
START_DATE	Start date must be mandatory.
	Start date must be later than the date of birth.
END_DATE	End date must be later than the start date
TYPE_OF_EMPLOYMENT	Type of employment, if not null, must be full time or part time.
FRACTION_OF_EMPLOYMENT	Fraction of employment, if not null, must be between 0.01 and 100.00
TENURE_OF_EMPLOYMENT	Tenure of employment if not null, must be CONTINUING, FIXED TERM, or CASUAL.
WEEKLY_WORK_HRS	Weekly work hours, if not null, must be between 0 and 168.

Table 3–17 Validations performed on data imported from the table IGS_AD_MILITARY_INT

Column Name	Validation performed
MILITARY_TYPE_CD	An active record must exist in IGS_LOOKUPS_VIEW for lookup code = field name and lookup type PE_MIL_SEV_TYPE.
ASSISTANCE_TYPE_CD	If not null, this must contain an active record in IGS_LOOKUPS_VIEW for lookup code = this field and lookup type = PE_MIL_ASS_TYPE.
ASSISTANCE_STATUS_CD	If not null, this must contain an active record in IGS_LOOKUPS_VIEW for lookup code = this field and lookup type = PE_MIL_ASS_STATUS.
START_DATE	Start date must be later than the date of birth.
END_DATE	If not null, this must be later than the start date.
SEPARATION_TYPE_CD	If not null, this must contain an active record in IGS_LOOKUPS_VIEW for lookup code = this field and lookup type = PE_MIL_SEP_TYPE.
Flexfield Columns, if provided	If value passed, then the descriptive flexfield, IGS_PE_IMMU_DTLS_FLEX must be configured.

Table 3–18 Validations performed on data imported from the table IGS_PE_IMMU_DTL_INT

Column Name	Validation performed
IMMUNIZATION_CODE	An active record must be present in IGS_LOOKUP_VALUES for lookup type PE_IMM_TYPE.
STATUS_CODE	An active record must be present in IGS_LOOKUP_VALUES for lookup type PE_IMM_STATUS.
START_DATE	Start date must be later than birth date.
START_DATE	Start date must be earlier than the end date.
Flexfield columns, if provided	If a value is passed, then the descriptive flexfield, IGS_PE_IMMU_DTLS_FLEX must be configured.

Table 3–19 Validations performed on data imported from the table IGS_AD_HLTH_INS_INT_ALL

Column Name	Validation performed
INSURANCE_CD	Must contain an active record in IGS_LOOKUPS_VIEW for lookup type PE_INS_TYPE
START_DATE	Start date must be later than date of birth.
END_DATE	If not null, this must be later than the start date.
Flexfield Columns, if provided	If value passed, then the descriptive flexfield, IGS_PE_HLTH_INS_ALL_FLEX must have been configured.

Table 3–20 Validations performed on data imported from the table IGS_PE_CITIZEN_INT

Column Name	Validation performed
COUNTRY_CODE	Validated from FND_TERRITORIES_VL.
DOCUMENT_TYPE	If not null, then an active record must be present in IGS_LOOKUPS_VIEW for lookup type CITIZENSHIP_DOC_TYPE.
DATE_RECOGNIZED	Date recognized must not be a null value.
	Date recognized must not be later than end date.
	Date recognized must not be earlier than the date of birth (if not null).
DATE_DISOWNED	Date disowned must not be earlier than date of birth (if not null).
	Date disowned must be between date recognized and end date.

Table 3–21 Validations performed on data imported from the table IGS_PE_FUND_SRC_INT

Column Name	Validation performed
FUND_SOURCE_CODE	An active record must be present in IGS_LOOKUPS_VIEW for lookup type PE_FUND_TYPE.
RELATIONSHIP_CODE	If not null, then an active record must be present in FND_LOOKUP_VALUES for lookup type PARTY_RELATIONS_TYPE.
DOCUMENT_IND	Must be either ‘Y’ or ‘N’.

Table 3–22 Validations performed on data imported from the table IGS_PE_VISA_INT

Column Name	Validation performed
[general]	If visa details are to be associated with a person’s passport details, the passport details must already exist in the system.
VISA_ISSUING_POST	Must match an active record in IGS_LOOKUP_VALUES for lookup type PE_US_VISA_ISSUE_LOC.
VISA_TYPE	Must be an active record in FND_LOOKUP_VALUES_VL for lookup type PER_US_VISA_TYPES.
VISA_ISSUE_DATE, VISA_EXPIRY_DATE	Visa issue date must be earlier than the visa expiry date.
VISA_ISSUE_DATE	Visa issue date must be later than the date of birth.
PASSPORT_ID	Passport ID for the person must match with his record in IGS_PE_PASSPORT.
AGENT_ORG_UNIT_CD	If the DYNAMIC SQL could not fetch any record.

Table 3–22 Validations performed on data imported from the table IGS_PE_VISA_INT

Column Name	Validation performed
AGENT_PERSON_ID	If not null, this must match a person record in IGS_PE_PERSON_BASE_V.
VISA_ISSUING_COUNTRY	If not null and not US, this must match an active record in FND_LOOKUP_VALUES for lookup type PER_US_COUNTRY_CODE.
VISA_ISSUING_COUNTRY, VISA_ISSUING_POST	This combination must be from an active record in IGS_LOOKUP_VALUES for lookup type PE_US_VISA_ISSUE_LOC.
Flexfield Columns, if provided	If a value is passed, then the descriptive flexfield, IGS_PE_INTL_VISA_FLEX must have been configured.
PORT_OF_ENTRY	Must match an active record in IGS_LOOKUP_VALUES for lookup type PE_US_PORT_OF_ENTRY.
VISIT_START_DATE, VISIT_END_DATE	Visit start date must be earlier than the visit end date.
VISIT_START_DATE	Visit start date must be later than the date of birth.
VISIT_END_DATE	If not null, this must be later than the date of birth.
VISIT_START_DATE	Visit start date must fall between issue date and expiry date.
VISIT_END_DATE	If not null, this must fall between the issue date and expiry date (with an extension of 30 days).

Table 3–23 Validations performed on data imported from the table IGS_PE_PASSPORT_INT

Column Name	Validation performed
PASSPORT_CNTRY_CODE	Must match an active record in FND_LOOKUP_VALUES_VL for lookup type PER_US_COUNTRY_CODE.
PASSPORT_EXPIRY_DATE	If birth date is not null, PASSPORT_EXPIRY_DATE must be later than the date of birth.

Table 3–24 Validations performed on data imported from the table IGS_PE_EIT_INT

Column Name	Validation performed
PEI_INFORMATION1	If not null, this must match a record in FND_LOOKUP_VALUES_VL for lookup type PER_US_COUNTRY_CODE.
START_DATE, END_DATE	Start date must be earlier than the end date.
END_DATE	End date must not be earlier than the date of birth.

Table 3–24 Validations performed on data imported from the table IGS_PE_EIT_INT

Column Name	Validation performed
START_DATE, END_DATE	Start and end dates must not overlap.
START_DATE	Start date must be later than the date of birth (if not null).
INFORMATION_TYPE	If not null and has a value PE_STAT_RES_COUNTRY, then this must be from an active record in FND_TERRITORIES_VL.
INFORMATION_TYPE	If not null and has a value PE_STAT_RES_STATE, then this must match a record in AR_LOC_ONE_LEVEL_V.
INFORMATION_TYPE	If not null and has a value PE_STAT_RES_STATUS, then must be from an active record in IGS_LOOKUPS_VIEW for lookup type PE_CITI_STATUS.
START_DATE, END _DATE	Start date must be earlier than the end date.

Table 3–25 Validations performed on data imported from the table IGS_PE_HOUSING_INT

Column Name	Validation performed
TEACH_PERIOD_RESID_ STAT_CD	Must match an active record in IGS_LOOKUP_VALUES for lookup type PE_TEA_PER_RES.
START_DATE	Start date must not be null.
START_DATE, END_DATE	If both are not null, the start date must be earlier than the end date.
CAL_TYPE, SEQUENCE_NUMBER	Both must be either null or not null.
CAL_TYPE, SEQUENCE_NUMBER	Person Housing Status Validation.A record matching calendar type and sequence must exist in IGS_CA_INST, IGS_EN_SU_ATTEMPT.

Table 3–26 Validations performed on data imported from the table IGS_PE_FLNY_DTL_INT

Column Name	Validation performed
CONVICT_IND	Must be either ‘Y’ or ‘N’.
CRIME_DATE	Must not be later than the system date.

Table 3–27 Validations performed on data imported from the table IGS_PE_HEAR_DTL_INT

Column Name	Validation performed
DSPL_FILE_IND	Must be either ‘Y’ or ‘N’.
ACAD_DISM_IND	Must be either ‘Y’ or ‘N’.
NON_ACAD_DISM_IND	Must be either ‘Y’ or ‘N’.
START_DATE, END_DATE	If Start Date and End date are not null then Start date must be earlier than the end date.
START_DATE	Start date must not be earlier than the date of birth.
START_DATE, END_DATE	If start date is null, end date must be also null.

Table 3–28 Validation performed on data imported from the table IGS_PE_RACE_INT

Column Name	Validation performed
RACE_CD	Must match an active record in IGS_LOOKUP_VALUES for lookup type PE_RACE.

Table 3–29 Validations performed on data imported from the table IGS_AD_CONTACT_INT_ALL

Column Name	Validation performed
CONTACT_POINT_TYPE	Must be an active record in FND_LOOKUP_VALUES for lookup type COMMUNICATION_TYPE and lookup code = EMAIL or PHONE.
PRIMARY_FLAG	If not null, this must be either ‘Y’ or ‘N’.
CONTACT_POINT_TYPE, PHONE_NUMBER AND PHONE_LINE_TYPE	If CONTACT_POINT_TYPE = ‘PHONE’ then PHONE_NUMBER and PHONE_LINE_TYPE must not be null.
PHONE_LINE_TYPE	Must be an active record in FND_LOOKUP_VALUES for lookup type PHONE_LINE_TYPE.
PHONE_COUNTRY_CODE	If not null, this must be present in HZ_PHONE_COUNTRY_CODES .
CONTACT_POINT_TYPE, EMAIL_ADDRESS	If contact point type is EMAIL, then email address must not be null.
EMAIL_FORMAT	Must be an active record in FND_LOOKUP_VALUES for lookup type = EMAIL_FORMAT.
LANGUAGE_NAME	Must be a active record in the FND_LANGUAGES_VL.

Table 3–30 Validations performed on data imported from the table IGS_AD_ACADHONOR_INT_ALL

Column Name	Validation performed
ACAD_HONOR_TYPE	Must be an active record in IGS_LOOKUP_VALUES for lookup type PE_ACAD_HONORS, If honor date is not null then it must be earlier than current system date and later than the birth date.
HONOR_DATE	If ACAD_HONOR_TYPE is valid and the honor date is not null, then it must be later than the birth date and earlier than the system date.
ACAD_HONOR_TYPE	Must be an active record in IGS_LOOKUP_VALUES for lookup type = PE_ACAD_HONORS.

Table 3–31 Validations performed on data imported from the table IGS_PE_RES_DTLS_INT

Column Name	Validation performed
RESIDENCY_CLASS_CD	Must be an active record in IGS_LOOKUP_VALUES for lookup type = PE_RES_CLASS.
RESIDENCY_STATUS_CD	Must be an active record in IGS_LOOKUP_VALUES for lookup type = PE_RES_STATUS.
START_DT	Start date must not be null.
	Start date must be later than the date of birth.
EVALUATION_DATE	Evaluation date must be later than the system date.
START_DT, END_DT	Start date must be earlier than the end date.
Flexfield columns, if provided	If value passed, then the descriptive flexfield, IGS_PE_PERS_RESIDENCY_FLEX must have been configured.
START_DT, END_DT	Date overlap check for start date and end date.

Table 3–32 Validations performed on data imported from the table IGS_AD_DISABLTY_INT_ALL

Column Name	Validation performed
DISABILITY_TYPE	Must match a record in IGS_AD_DISBL_TYPE.
	If DISABILITY_TYPE is NONE, then no disability must be present for the person.
	If DISABILITY_TYPE is not NONE, then a disability must be present for the person.
ELIG_EARLY_REG_IND	ELIG_EARLY_REG_IND must be either ‘Y’ or ‘N’.
SPECIAL_ALLOW_CD	If not null, this must match an active record in IGS_LOOKUP_VALUES for lookup type PE_SN_ALLOW.

Table 3–32 Validations performed on data imported from the table IGS_AD_DISABLTY_INT_ALL

Column Name	Validation performed
SUPPORT_LEVEL_CD	If not null, then this must match an active record in IGS_LOOKUP_VALUES for lookup type PE_SN_ADD_SUP_LVL.
START_DATE, END_DATE	Start date must be earlier than the end date. If the end date is not null, then it must be earlier than the system date.
START_DATE	Start date must be later than the date of birth.
INTERVIEWER_DATE	If not null, this must be later than the date of birth.
INTERVIEWER_ID	Interviewer ID must be different from the person ID.
	Interviewer ID must be a valid person ID.

Table 3–33 Validations performed on data imported from the table IGS_PE_SN_SRVCE_INT

Column Name	Validation performed
SPECIAL_SERVICE_CD	Must match an active record in IGS_LOOKUPS_VIEW for lookup type PE_SN_SERVICE.
DOCUMENTED_IND	Must be either ‘Y’ or ‘N’.
START_DT, END_DT	If start date is null, end date must be null.
	Start date must be earlier than the end date.
START_DT	Start date must be later than the date of birth.

Table 3–34 Validation performed on data imported from the table IGS_PE_SN_CONCT_INT

Column Name	Validation performed
CONTACT_DATE	If not null, this must be later than the date of birth.

Table 3–35 Validations performed on data imported from the table IGS_AD_RELATIONS_INT_ALL

Column Name	Validation performed
RELATIONSHIP_CODE	RELATIONSHIP_CODE must not be null.
PRE_NAME_ADJUNCT	If not null, then this must be an active record in FND_LOOKUP_VALUES for lookup type = CONTACT_TITLE.
RELATIONSHIP_TYPE	RELATIONSHIP_TYPE must not be null and a lookup must exist in FND_LOOKUP_VALUES lookup type PARTY_RELATIONS_TYPE for a PERSON-PERSON relationship.

Table 3–35 Validations performed on data imported from the table IGS_AD_RELATIONS_INT_ALL

Column Name	Validation performed
OTH_RELATIONSHIP_TYPE	If not null, this must be a lookup in FND_LOOKUP_VALUES lookup type PARTY_RELATIONS_TYPE.
PERSON_NUMBER	PERSON_NUMBER must not be null, when the profile HZ_GENERATE_PARTY_NUMBER is set to N.
SEX	If not null then Sex must be 'M' or 'F'.
BIRTH_DT	If not null, this must not be later than the system date.
DECEASED_DATE	If deceased = 'Y' then DECEASED_DATE must not be later than the system date.

Table 3–36 Validations performed on data imported from the table IGS_AD_RELADDR_INT_ALL

Column Name	Validation performed
SITE_USE_CODE_1	If not null, then this must be an active record in FND_LOOKUP_VALUES for lookup type=' PARTY_SITE_USE_CODE'
SITE_USE_CODE_2	If not null, then this must be an active record in FND_LOOKUP_VALUES for lookup type=' PARTY_SITE_USE_CODE'
SITE_USE_CODE_1 *and* SITE_USE_CODE_2	These two fields must not be the same.
COUNTRY	Must be an active record in FND_TERRITORIES_VL.
CORRESPONDENCE	Must be either 'Y' or 'N'.

Table 3–37 Validations performed on data imported from the table IGS_AD_REL_CON_INT_ALL

Column Name	Validation performed
CONTACT_POINT_TYPE	Must be an active record in FND_LOOKUP_VALUES for lookup type = COMMUNICATION_TYPE and lookup code in 'EMAIL','PHONE'.
PRIMARY_FLAG	If not null, then this must be either 'Y' or 'N'.
CONTACT_POINT_TYPE, PHONE_NUMBER *and* PHONE_LINE_TYPE	If contact point type = 'PHONE' then both PHONE_NUMBER and PHONE_LINE_TYPE must not be null.
PHONE_LINE_TYPE	If contact point type = 'PHONE' then PHONE_LINE_TYPE must be from an active record in FND_LOOKUP_VALUES for lookup type = PHONE_LINE_TYPE.

Table 3–37 Validations performed on data imported from the table IGS_AD_REL_CON_INT_ALL

Column Name	Validation performed
PHONE_COUNTRY_CODE	If not null, then this must be a value in HZ_PHONE_COUNTRY_CODES.
CONTACT_POINT_TYPE *and* EMAIL_ADDRRESS	If contact point type = 'EMAIL' then email address must not be null.
EMAIL_FORMAT	Must be an active record in FND_LOOKUP_VALUES for lookup type = EMAIL_FORMAT

Table 3–38 Validations performed on data imported from the table IGS_AD_RELACAD_INT_ALL

Column Name	Validation performed
INSTITUTION_CODE	Must be a valid IGS_OR_INSTITUTION based on active INSTITUTION_STATUS selected from IGS_OR_INST_STAT.
PROGRAM_TYPE_ATTEMPTED	If not null, this must be a record in IGS_PS_TYPE.
PROGRAM_TYPE_EARNED	If not null, this must be a record in IGS_PS_TYPE.
PROGRAM_CODE, VERSION_NUMBER	Both must be either null or both must be not null.
	If both are not null, then the combination must be in IGS_PS_VER.
CURRENT_INST	Must be either 'Y' or 'N'.
CURRENT_INST, END_DATE	If CURRENT_INST = 'Y' then end date must be null.
START_DATE	If not null, this must be earlier than current system date.
END_DATE, START_DATE	If both are not null, then end date must be later than start date.
END_DATE, START_DATE	If start date is null, then end date must also be null.
PLAN_COMPLETION_DATE, START_DATE	If both not null, then plan completion date must be later than the start date.
START_DATE, BIRTH_DATE	Start date must be later than the date of birth.

Table 3–39 Validations performed on data imported from the table IGS_AD_RELEMP_INT_ALL

Column Name	Validation performed
OCCUPATIONAL_TITLE_CODE	If not null, this is validated from IGS_PS_DIC_OCC_TITLS.
START_DT	Must not be null.
START_DATE, END_DT	End date must be later than start date.
TYPE_OF_EMPLOYMENT	Type of employment must be either full time or part time.
FRACTION_OF_EMPLOYMENT	Must be between 0.01 and 100.00
TENURE_OF_EMPLOYMENT	Must be CONTINUING, FIXED TERM or CASUAL.
WEEKLY_WORK_HOURS	Must be between 0 and 168.

4

Institutions Import Process

This chapter describes the process of importing institution data. The following sections are in this chapter:

Definition

This chapter describes the process of importing data about institutions into Oracle Student System.

Overview

Data about institutions, prospective student, and applicants is required for the admission process and can also be a useful tool in recruitment of students. For example, institutions send transcripts of applicants' coursework and grades as part of the regular admissions process. The name, location, and other contact information about the institution is used in the applicant admission process.

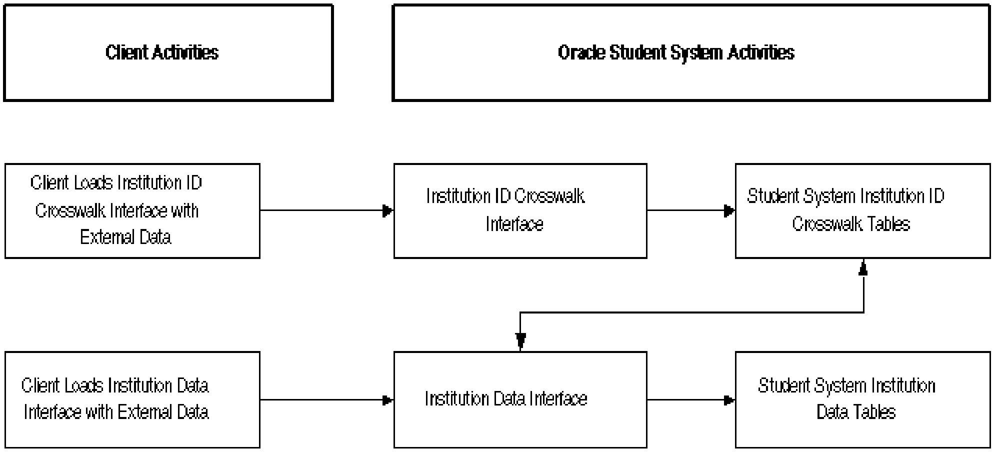
Additional information about institutions can also be used in the student recruiting process. For example, information such as the percentage of students that take advanced course work, or the percentage that take the College Boards, can be used to focus a client's recruiting efforts on schools that have the highest numbers of students of the caliber that the client would like to attract.

Information about prospective students' interests and special requirements can also be used to target students whose personal profiles match a client's target population. By accumulating this data within Oracle Student System, clients can perform reporting, modeling, and forecasting that can increase the effectiveness of their admissions functions and their recruiting efforts.

Import Institutions Process Flow Diagram

Figure 4–1, page 4-2 shows the import institutions process flow diagram as described in the accompanying text.

Figure 4–1 Import Institutions Process Flow Diagram


Client Activities
Oracle Student System Activities
Client Loads Institution ID Crosswalk Interface with External Data
Institution ID Crosswalk Interface
Student System Institution ID Crosswalk Tables
Client Loads Institution Data Interface with External Data
Institution Data Interface
Student System Institution Data Tables

Institutions Import Process

1. Users load the interface crosswalk tables with external data.

   To enable Oracle Student System to correctly match and update data about the same institution originating from different sources, there must be an intervening crosswalk that translates the different institution IDs from the different sources to each other and to a common ID in the crosswalk. This common ID is auto-generated during the import of data from the Crosswalk

Interface to Oracle Student System Crosswalk. The ID is used to link institutions in the Crosswalk and is not involved in any subsequent processes in the import.

Data must be loaded into the crosswalk interface in the format specified by Oracle Student System. Populating the crosswalk interface requires system administrator skills using SQL loader or a similar tool.

2. The Import Institution Crosswalk concurrent program is run and the Oracle Student System Crosswalk is populated. The Crosswalk tables are used to identify existing institutions in the database.
3. Any errors generated during the Import Institution Crosswalk concurrent program can be viewed in the Import Crosswalk Error report. Users correct the Error record in the Interface Crosswalk tables and run the Import Institution Crosswalk concurrent program. Steps 2 and 3 are repeated until all records in the Interface Crosswalk table are imported to Oracle Student System Crosswalk tables. If an error occurs during the import process, the corresponding record is marked as an Error and the process proceeds to the next record. Users can correct the error, change the status of the record to Pending, and re-run the process.
4. The Import Institution concurrent program is run.

   The Import Institutions concurrent program is used to import institutions from Interface tables to Oracle Student System tables. Users populate the Interface tables with data from different sources, and the Import Institutions concurrent program takes data from these tables, validates and creates or updates the institution's details in the Oracle Student System tables.
5. If any errors are generated during the import of records from Interface tables to Oracle Student System tables, the institution record in the Interface table is marked as an Error and is not imported.

   Users can generate the following reports to check and correct the error records: Import Crosswalk Error Report and Import Institutions Error Report.
6. The Import Institution Error Report is run and error records in the Interface table are corrected. The record status is changed to Pending.
7. The Import Institution process is run again to import the corrected institutions.

   When the import process is run, each record is flagged indicating if the record is imported. Records are flagged as Successful or Error. All records remain in the interface until purged by the user. If the import process is re-run, all records flagged as Successful or Error are ignored and only new or corrected records are imported. Records previously imported are not re-imported.

Interface Tables

Oracle Student System uses the following interface tables during the institution import process:

- IGS_OR_ADR_INT
- IGS_OR_ADRUSGE_INT
- IGS_OR_CWLK
- IGS_OR_CWLK_DTL
- IGS_OR_CWLK_DTL_INT
- IGS_OR_CWLK_INT
- IGS_OR_INST_BTCH_INT
- IGS_OR_INST_CON_INT
- IGS_OR_INST_CPHN_INT
- IGS_OR_INST_INT
- IGS_OR_INST_NTS_INT
- IGS_OR_INST_SDTL_INT
- IGS_OR_INST_STAT_INT

For information on table column descriptions and values, see Oracle Student System Table Information at http://metalink.oracle.com/metalink/plsql/ml2_gui.startup.

5

Probability Values Import Process

This chapter describes the process of collecting and transferring probability values from the interface tables in the Admissions module of Oracle Student System to the target table. The following sections are in this chapter:

- Definition
- Overview
- Probability Values Import Process Flowchart
- Probability Values Import Process
- Interface Tables

Definition

The Probability Values Import process is a concurrent program that enables users to collect and transfer probability values from the interface tables in the Admissions module to the Oracle Student System target table.

Oracle Student System probability values represent the following:

- relative likelihood that a student will apply, enroll, or attend the institution given a specified recruitment effort

Overview

Users extract pertinent data from the Person Reference and Admissions modules in Oracle Student System using appropriate tools and employ third party software or a third party provider to analyze the extracted data and perform calculations that result in probability values. These probability values represent the relative likelihood that a student will apply, enroll, or attend the institution given a specified recruitment effort.

Once the results of the analysis and calculations have been received by users, they populate the IGS_AD_RECRT_PI_HDR_ALL and IGS_AD_RECRT_PI_INT_ALL interface tables with the probability values. Records populated into the IGS_AD_RECRT_PI_INT_ALL interface table are associated with a person record using Person ID. The IGS_AD_RECRT_PI_INT_ALL interface table has the primary key of Person ID.

To collect and transfer data from the IGS_AD_RECRT_PI_INT_ALL interface table to the Oracle Student System IGS_AD_RECRUIT_PI target table, users run the Probability Values Import process.

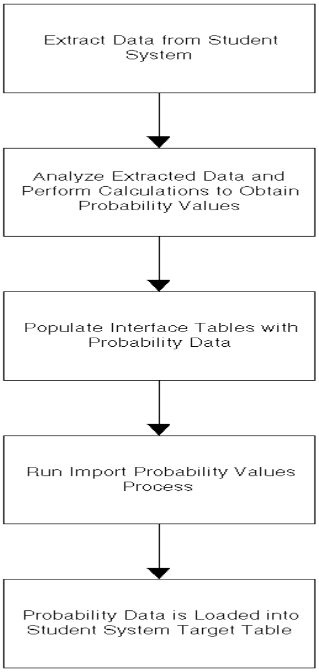
Reference

For additional information on the Probability Values Import process, see Chapter 4, Admissions, *Oracle Student System User's Guide*.

Probability Values Import Process Flowchart

Figure 5–1, page 5-3 shows the Probability Values Import process as described in the Probability Values Import Process section.

Figure 5–1 Probability Values Import Process Flowchart


Extract Data from Student System
Analyze Extracted Data and Perform Calculations to Obtain Probability Values
Populate Interface Tables with Probability Data
Run Import Probability Values Process
Probability Data is Loaded into Student System Target Table

Probability Values Import Process

The Probability Values Import process comprises the following steps:

1. Users extract pertinent data from the Person Reference and Admissions modules in Oracle Student System using Oracle Discoverer, SQL Plus, or third party tools like SQL Navigator.
2. Users employ third party software or a third party provider to analyze the extracted data and perform calculations that result in probability values.
3. Users populate the IGS_AD_RECRT_PI_HDR_ALL and IGS_AD_RECRT_PI_INT_ALL interface tables with the calculated probability values using third party software, such as SQL Loader or a custom P1/SQL routine.
4. Users must generate value PRBLTY_VAL_BATCH _ID from the IGS_AD_RECRT_PI_HDR_ALL interface table and value

PRBLTY_VAL_INT_ID from the IGS_AD_RECRT_PI_INT_ALL interface table using the corresponding sequence generators IGS_AD_RECRT_PI_HDR_S and IGS_AD_RECRT_PI_INT_S.

5. Users run the Probability Values Import process to collect and transfer the probability data from the IGS_AD_RECRUIT_PI_INT_ALL interface table to the the Oracle Student System IGS_AD_RECRUIT_PI target table.
6. During transfer of the probability data from the interface table to the Oracle Student System target table, the Probability Values Import process checks for duplicate records.
   - If a record is not a duplicate, it is inserted into the Oracle Student System IGS_AD_RECRUIT_PI target table.
   - If a record is a duplicate, the MATCH_IND is set to 23.
   - If a record is a duplicate and the MATCH_IND = 21, the record has been reviewed and the values are transferred from the interface table to the Oracle Student System target table.
   - The valid values for the MATCH_IND field are as follows:

     MATCH_IND = 21; record reviewed and ready for import

     MATCH_IND = 23; duplicate record found and ready for review

     Note: Users must review all records where MATCH_IND = 23.
7. If users want a particular record to override in the Oracle Student System target table with the interface table record values, they must change the MATCH_IND value from 23 to 21.
8. To process updated records, users must re-run the Probability Values Import process.
9. Once probability records are successfully imported into the Oracle Student System target tables, the RECORD_STATUS in the IGS_AD_RECRT_PI_INT_ALL interface table is updated to 1. All records with Status 1 are deleted from the IGS_AD_RECRT_PI_INT_ALL interface table.
10. The Probability Values Import process logs the following types of errors if records fail importation:
    - E001 = Validation failed; Probability Type Code Id

      Note: You must specify a valid Probability Type existing in igs_ad_code_classes, where class = 'PROB_TYPE'

- E002 = Validation failed; Probability Source Code Id

  Note: You must provide a valid Probability Type existing in igs_ad_code_classes, where class = 'PROB_SOURCE'

- E003 = Mandatory Probability Type Code Id not Populated
- E004 = Mandatory Calculation Date not Populated
- E005 = Duplicate Record Found
- E006 = Insert Failed
- E007 = Update Failed

11. Users can correct errors directly in the IGS_AD_RECRT_PI_INT_ALL interface table using SQL Plus or third party tools, such as SQL Navigator.
12. To process updated records, users must re-run the Probability Values Import process.

Interface Tables

Oracle Student System uses the following interface tables during the Probability Values Import process.

- IGS_AD_RECRT_PI_HDR_ALL
- IGS_AD_RECRT_PI_INT_ALL

For information on table column descriptions and values, see Oracle Student System Table Information at http://metalink.oracle.com/metalink/plsql/ml2_gui.startup.

6

Academic Index Import Process

This chapter describes the Academic Index Import process. The following sections are in this chapter:

Definition

The Academic Index Import process is a concurrent program that enables users to collect and transfer index data from interface tables to Oracle Student System target tables in the Admissions module.

Overview

Colleges and universities use various methods to determine whether to admit or deny admission to students. One method involves the calculation and use of an academic index, which is used to evaluate students. Using mathematical models that include variables such as test scores, GPA, and rank in class, users develop indices and formulae that predict college success and assist in automated admission decisions.

Using user-defined and stored algorithms or third party calculators, users calculate academic index values for each admission application by applying academic index calculation rules to satisfy their business requirements. Users extract pertinent

Oracle Student System data and then employ third party tools to populate the open interface table with the calculated academic index values. Users run the Academic Index Import process to collect and transfer the academic index data from the interface table to the Oracle Student System target tables.

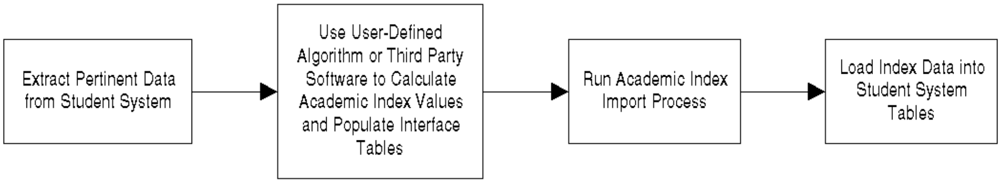
Reference

For additional information on the Academic Index Import process, see Chapter 4, Admissions, Oracle Student System User's Guide.

Academic Index Import Process Flowchart

Figure 6–1, page 6-2 shows the Academic Index Import process as described in the Academic Index Import Process section.

Figure 6–1 Academic Index Import Process Flowchart


Extract Pertinent Data from Student System
Use User-Defined Algorithm or Third Party Software to Calculate Academic Index Values and Populate Interface Tables
Run Academic Index Import Process
Load Index Data into Student System Tables

Academic Index Import Process

The Academic Index Import process comprises the following steps:

1. Users extract relevant Oracle Student System data.
2. Users employ a user-defined and stored algorithm or third party software to calculate the academic index values and to return the results to the IGS_AD_ACIDX_HDR_ALL and IGS_AD_ACAD_IDX_INT interface tables. The group of academic index values for admission applications is stored in the IGS_AD_ACAD_IDX_INT interface table.

   A group of records populated into the IGS_AD_ACAD_IDX_INT open interface table has a Batch ID. The Batch ID is recorded in the IGS_AD_ACIDX_HDR_ALL table. Records that are populated into the IGS_AD_ACAD_IDX_INT interface table are given a status of Pending.

3. Users run the Academic Index Import process for a particular Batch ID, which is passed as a parameter from the concurrent manager. The Academic Index Import process retrieves the academic index records from the IGS_AD_ACIDX_HDR_ALL and IGS_AD_ACAD_IDX_INT interface tables whose status is Pending.
4. The Academic Index Import process processes all rows in the IGS_AD_ACAD_IDX_INT interface table for the assigned batch ID and imports the data into the Oracle Student System IGS_AD_PS_APPL_INST_ALL target table.
5. The Oracle Student System tables are updated only when the Documentation Status = Complete and the Admission Outcome Status = Pending.
6. During the import of index data from the IGS_AD_ACAD_IDX_INT interface table into the Oracle Student System target table, a record is updated as an Error in the Status column and the Admissions Import process proceeds to the next record in the IGS_AD_ACAD_IDX_INT interface table if the following error conditions occur:
   - no corresponding record found in the Oracle Student System target table for the given record in the interface table
   - corresponding record is locked by another user
   - documentation status of the corresponding application is not mapped to the system documentation status Satisfied and Admission Outcome Status is not Pending
7. If an error occurs when transferring index data from the IGS_AD_ACAD_IDX_INT interface table to the Oracle Student System target table, corrections must be made directly in the IGS_AD_ACAD_IDX_INT interface table.
8. After correcting errors, users must change the record status to Pending and re-run the Academic Index Import process.
9. Index records that successfully transfer to the Oracle Student System target table are purged from the IGS_AD_ACIDX_HDR_ALL and IGS_AD_ACAD_IDX_INT interface tables.

Interface Tables

Oracle Student System uses the following interface tables during the Academic Index Import process.

- IGS_AD_ACIDX_HDR_ALL
- IGS_AD_ACAD_IDX_INT_ALL

For information on table column descriptions and values, see Oracle Student System Table Information at http://metalink.oracle.com/metalink/plsql/ml2_gui.startup.

7

Test Score Conversion and Import Process

This chapter describes the process of receiving test scores from external testing agencies, converting and transferring them to interface tables, and then running a concurrent program to collect and transfer the test scores from the interface tables to Oracle Student System target tables. The following sections are in this chapter:

- Definition
- Overview
- Prerequisites
- Manual Test Score Conversion and Import Process Flowchart
- Manual Test Score Conversion and Import Process
- Batch Test Score Conversion and Import Process Flowchart
- Batch Test Score Conversion and Import Process
- Interface Tables

Definition

The Test Score Conversion and Import process enables users to convert and transfer test scores individually or as a batch to interface tables, and then run a concurrent program to collect and transfer the test scores from the interface tables to Oracle Student System target tables.

Overview

Admissions personnel at institutions receive test scores for students from testing agencies. Test scores are received in paper form or in electronic form. Paper scores must be entered manually into Oracle Student System. Electronic files must be loaded using the import process, matched against person ID records, or stored for future use.

Test score conversion is a common practice when an institution prefers one particular test score over another. For example, suppose a university prefers SAT scores over ACT scores. Institutions use concordance tables provided by one or more testing agencies to convert different test scores to one uniform level. For example, ACT test scores can be converted into equivalent SAT scores. The values provided in the table can be populated in the user hook for test score conversion.

Institutions may also use some combination of test scores to produce a decision score. For example, a decision score might be the highest combined score on a single test, the best verbal and math score from any number of official scores, or a weighted average of all verbal and math scores from an SAT or GRE.

To convert test scores for an individual, users navigate to the Admission Test Results window and click **Convert Test Scores**, which calls the user hook procedure. The user hook is an institution-developed and coded software procedure that converts, combines, or calculates test scores.

Alternatively, to begin a batch conversion of test scores, users run the Convert and Import Test Scores process, a concurrent program, which converts the test scores for all the persons included in the Person Group ID passed as the parameter.

The Convert and Import Test Scores process calls the user hook procedure. The user hook procedure, written by the institution, evaluates the test scores and populates the IGS_AD_TSTRST_UH_INT and IGS_AD_TSTDTL_UH_INT interface tables with the converted test scores. The concurrent program processes all the rows in the interface tables, along with the same unique identifier that was passed to it by the user hook.

The converted test scores are collected from the interface tables and transferred to Oracle Student System Admission target tables for the Person ID and Test Type associated with the row of the process.

Reference

For information on importing test scores without conversion, see Oracle Student System Admission Import Process, page 3-1.

For additional information on the Test Score Conversion and Import process, see Chapter 4, Admissions, Oracle Student System User’s Guide.

Prerequisites

Before running the Manual Test Score Conversion and Import process, users must perform the following steps:

❏ Define test types and test segments in the Admission Test Types window.

❏ For details on prerequisite steps to perform before running the Manual Test Score Conversion and Import process, see Chapter 4, Admissions, Oracle Student System User’s Guide.

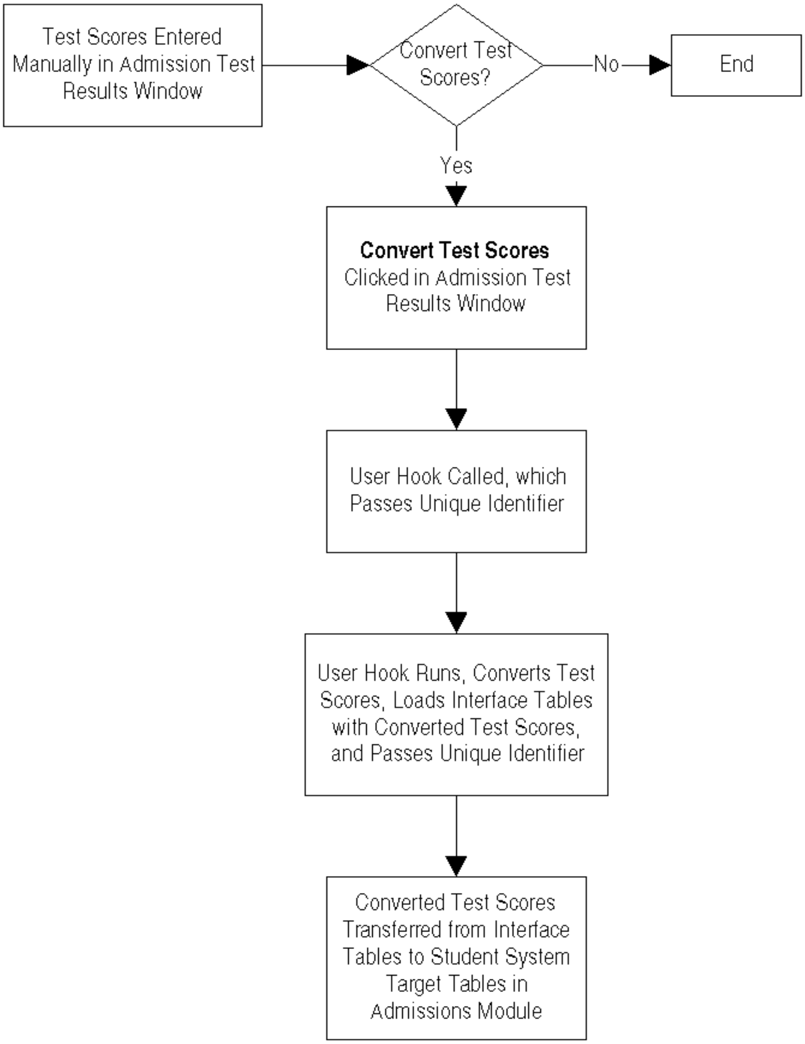
Before running the Batch Test Score Conversion and Import process, users must perform the following steps:

❏ For details on prerequisite steps to perform before running the Batch Test Score Conversion and Import process, see Chapter 4, Admissions, Oracle Student System User’s Guide.

Manual Test Score Conversion and Import Process Flowchart

Figure 7–1, page 7-4 shows the Manual Test Score Conversion and Import process as described in the Manual Test Score Conversion and Import Process section.

Figure 7–1 Manual Test Score Conversion and Import Process Flowchart


Test Scores Entered Manually in Admission Test Results Window
Convert Test Scores?
No
End
Yes
Convert Test Scores Clicked in Admission Test Results Window
User Hook Called, which Passes Unique Identifier
User Hook Runs, Converts Test Scores, Loads Interface Tables with Converted Test Scores, and Passes Unique Identifier
Converted Test Scores Transferred from Interface Tables to Student System Target Tables in Admissions Module

Manual Test Score Conversion and Import Process

The manual test score conversion and import process comprises the following steps:

1. In Oracle Student System, navigate to the Admission Test Results window as follows:

   Admission - Direct Admission

   The Direct Admission window appears.

2. Enter or query the appropriate record.
3. Click **More** in the header region of the Direct Admission window.
4. Click **Admission Tests**.

   The Admission Test Summary window appears.

5. Click **New** or **Open**.

   The Admission Test Results window appears.

6. To call the user hook procedure, IGS_AD_TSTUH_CALL_PKG.CALL_USER_HOOK, with the parameter PERSON_ID and the out parameter L_SESSION_ID, click **Convert Test Scores**.
7. The user hook procedure, IGS_AD_TSTUH_CALL_PKG.CALL_USER_HOOK, generates a unique SESSION_ID from the sequence, which in turn calls the Exact User Hook procedure with the parameters PERSON_ID and SESSION_ID.

   The Exact User Hook procedure is called from inside the package IGS_AD_TSTUH_CALL_PKG.

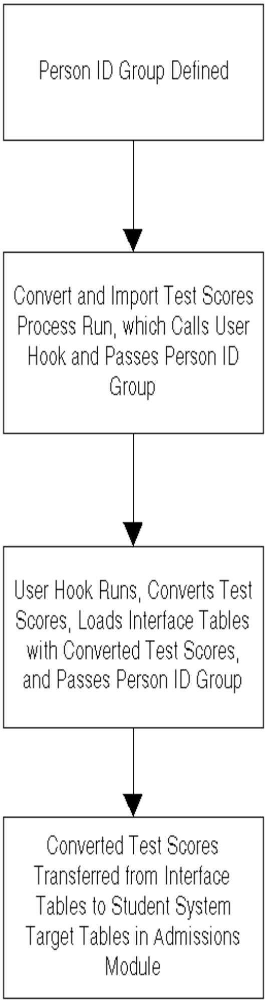
8. The user hook procedure logic written by the institution converts test scores and populates the IGS_AD_TSTRST_UH_INT and IGS_AD_TSTDTL_UH_INT interface tables with calculated test types, test segments, and test scores.
9. Users must change the STATUS of records inserted into the interface tables to 2 = Pending.
10. After the Exact User Hook procedure is executed, the IGS_AD_IMP_UH_TST_PKG.TRANSFER_INT_OSS procedure is invoked with the parameter PERSON_ID and the out parameter L_SESSION_ID. The IGS_AD_IMP_UH_TST_PKG.TRANSFER_INT_OSS procedure collects and transfers data from interface tables to Oracle Student System target tables.

11. If the transfer of data from the interface tables to the Oracle Student System tables fails, the pertinent records are marked with Error Code E002. This error appears in the log file as Insertion of Test Result Record Failed.
12. Records that successfully transfer to Oracle Student System target tables are purged from the interface tables.

Batch Test Score Conversion and Import Process Flowchart

Figure 7–2, page 7-7 shows the Batch Test Score Conversion and Import process as described in the Batch Test Score Conversion and Import Process section.

Figure 7–2 Batch Test Score Conversion and Import Process Flowchart


Person ID Group Defined
Convert and Import Test Scores Process Run, which Calls User Hook and Passes Person ID Group
User Hook Runs, Converts Test Scores, Loads Interface Tables with Converted Test Scores, and Passes Person ID Group
Converted Test Scores Transferred from Interface Tables to Student System Target Tables in Admissions Module

Batch Test Score Conversion and Import Process

The Batch Test Score Conversion and Import process is comprised of the following steps:

1. Users define a Person ID Group.

   To define Person ID Groups, see Chapter 3, Person, Oracle Student System User’s Guide.

2. Users run the Convert and Import Test Scores process, a concurrent program, which converts test scores for all persons included in the Person Group ID passed as the parameter.

3. The Convert and Import Test Scores process takes all the persons associated with the Person Group ID and calls the user hook procedure, IGS_AD_TSTUH_CALL_PKG.CALL_USER_HOOK, with the parameter PERSON_ID and the out parameter L_SESSION_ID.

4. The user hook procedure, IGS_AD_TSTUH_CALL_PKG.CALL_USER_HOOK, generates a unique SESSION_ID from the sequence, which in turn calls the Exact User Hook procedure with the parameters PERSON_ID and SESSION_ID. The Exact User Hook procedure is called from inside the package IGS_AD_TSTUH_CALL_PKG.

5. The user hook procedure logic written by the institution converts test scores and populates the IGS_AD_TSTRST_UH_INT and IGS_AD_TSTDTL_UH_INT interface tables with calculated test types, test segments, and test scores.

6. Users must change the STATUS of records inserted into the interface tables to 2 = Pending.

7. After the Exact User Hook procedure is executed, the IGS_AD_IMP_UH_TST_PKG.TRANSFER_INT_OSS procedure is invoked with the parameter PERSON_ID and the out parameter L_SESSION_ID. The IGS_AD_IMP_UH_TST_PKG.TRANSFER_INT_OSS procedure collects and transfers data from the interface tables to the Oracle Student System target tables.

8. If the transfer of data from the interface tables to the Oracle Student System target tables fails, the pertinent records are marked with Error Code E002. This error appears in the log file as Insertion of Test Result Record Failed.

9. Records that successfully transfer to the Oracle Student System target tables are purged from the interface tables.

10. Repeat Steps 2 to 7 for all Person ID Groups.

Interface Tables

Oracle Student System uses the following interface tables during the Manual or Batch Test Score Conversion and Import process.

- IGS_AD_TSTRST_UH_INT
- IGS_AD_TSTDTL_UH_INT

For information on table column descriptions and values, see Oracle Student System Table Information at http://metalink.oracle.com/metalink/plsql/ml2_gui.startup.

8

Automatic Batch Admission Decisions Import Process

This chapter describes the process of automating batch admission decisions based on user-defined rules, indices, and formulas. The following sections are in this chapter:

- Definition
- Overview
- Automatic Batch Admission Decisions Import Process Functional Flowchart
- Automatic Batch Admission Decisions Import Process
- Interface Tables

Definition

The Automatic Batch Admission Decisions Import process is an internal process that enables users to collect and transfer batch admission decisions from interface tables to Oracle Student System target tables.

Overview

Many institutions use automated decision-making for admission decisions, where offers are made or admissions denied based on a formulaic review of criteria.

To begin the Automatic Batch Admission Decisions Import process, users extract application data from the IGS_AD_PS_APPL_INST_ALL table in the Admissions module using Discoverer or any third party software and load the application data into interface tables.

Users then navigate to the Admission Decision Details window to specify application decisions, select specified applications, and decide whether to post or discard admission decisions based on index-based or rule-based criteria.

If users post admission decisions, they click **Post Decisions** in the Admission Decision Details window and the Automatic Batch Admission Decisions Import process collects and transfers the application decisions from the interface tables to the Oracle Student System table.

Once users post admission decisions, the admission outcome status, which is the progress of the applicant's request for admission, and decision details of the selected batch records are updated in the Oracle Student System IGS_AD_PS_APPL_INST_ALL table.

If users discard admission decisions, they click **Discard Decisions** in the Admission Decision Details window and the details of the selected batch records are deleted from the interface tables.

References

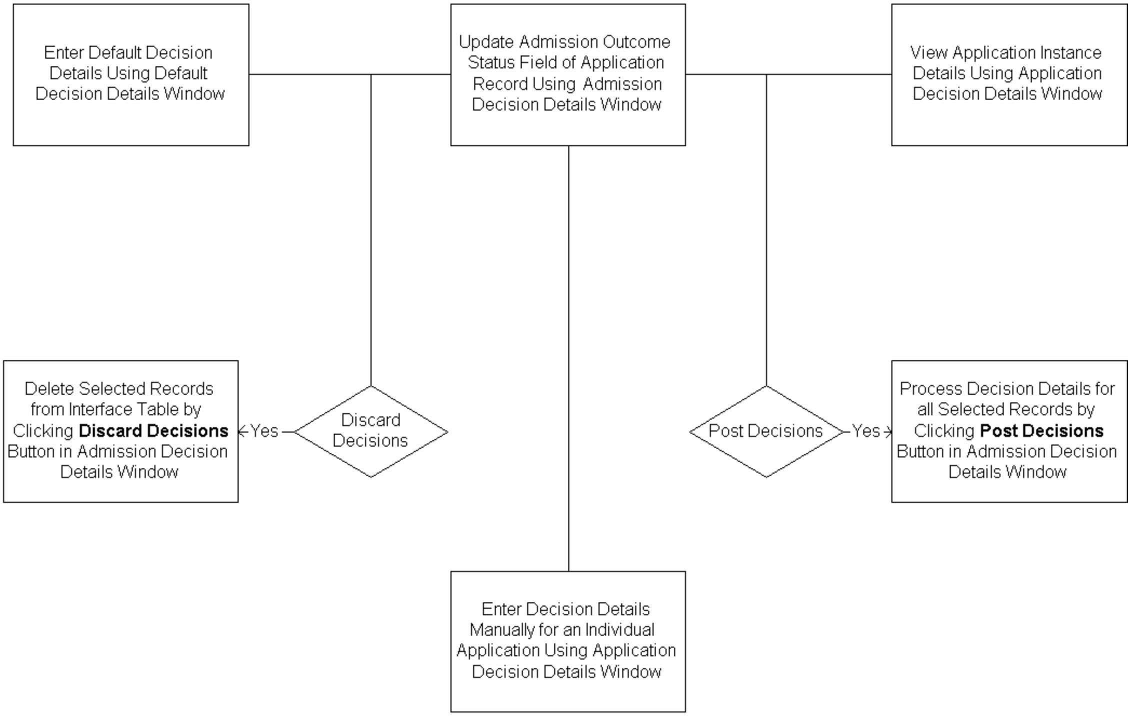
For information on using the Admission Decision Details window, see Chapter 4, Admissions, *Oracle Student System User's Guide*.

For information on using the Application Decision Details window, see Chapter 4, Admissions, *Oracle Student System User's Guide*.

Automatic Batch Admission Decisions Import Process Functional Flowchart

Figure 8–1, page 8-3 shows the Automatic Batch Admission Decisions Import process from a functional perspective and indicates windows users open as described in the accompanying text.

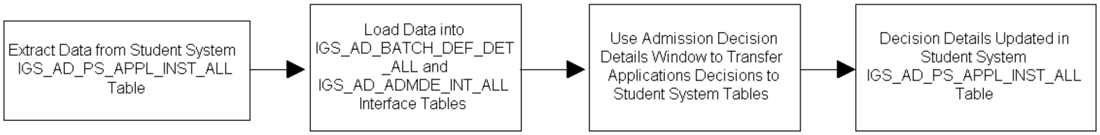
Figure 8–1 Automatic Batch Admission Decisions Import Process Functional Flowchart


Enter Default Decision Details Using Default Decision Details Window
Update Admission Outcome Status Field of Application Record Using Admission Decision Details Window
View Application Instance Details Using Application Decision Details Window
Delete Selected Records from Interface Table by Clicking **Discard Decisions** Button in Admission Decision Details Window
Yes
Discard Decisions
Post Decisions
Yes
Process Decision Details for all Selected Records by Clicking **Post Decisions** Button in Admission Decision Details Window
Enter Decision Details Manually for an Individual Application Using Application Decision Details Window

Automatic Batch Admission Decisions Import Process Flowchart

Figure 8–2, page 8-4 shows the Automatic Batch Admission Decisions Import process from a technical event perspective as described in the Automatic Batch Admission Decisions Import Process section.

Figure 8–2 Automatic Batch Admission Decisions Import Process Flowchart


Extract Data from Student System IGS_AD_PS_APPL_INST_ALL Table
Load Data into IGS_AD_BATCH_DEF_DET_ALL and IGS_AD_ADMDE_INT_ALL Interface Tables
Use Admission Decision Details Window to Transfer Applications Decisions to Student System Tables
Decision Details Updated in Student System IGS_AD_PS_APPL_INST_ALL Table

Automatic Batch Admission Decisions Import Process

The Automatic Batch Admission Decisions Import process comprises the following steps.

1. Users extract application data from IGS_AD_PS_APPL_INST_ALL using Discoverer or any third party software and load the data into the IGS_AD_BATCH_DEF_DET_ALL and IGS_AD_ADMDE_INT_ALL interface tables.
2. In Oracle Student System, users navigate to the Admission Decision Details window as follows:

   Oracle Student System - Admission - Admission Decision Details
3. In the Admission Decision Details window, users specify admission outcome status, select specified application records, and decide whether to post or discard admission decisions by clicking **Post Decisions** or **Discard Decisions**.
4. Only records with the following statuses are selected for the automatic batch admission decisions import process:
   - Admission Outcome Status = Pending
   - Documentation Status = Complete
   - Qualification Status = Qualified
5. If users post admission decisions by clicking **Post Decisions** in the Admission Decision Details window, the Automatic Batch Admission Decisions Import process collects and transfers the application decisions from the IGS_AD_BATCH_DEF_DET_ALL and IGS_AD_ADMDE_INT_ALL interface tables to the Oracle Student System target table. The admission outcome status and decision details of the selected batch records are updated in the Oracle Student System IGS_AD_PS_APPL_INST_ALL target table.

6. If users discard admission decisions by clicking **Discard Decisions** in the Admission Decision Details window, the details of the selected batch records are deleted from the interface tables.
7. If an error occurs when transferring application data from the interface tables to the Oracle Student System target table, corrections must be made directly in the interface tables or in the Oracle Student System target table, depending on the source of the error. After the error is corrected, users must navigate to the Admission Decision Details window, select specified applications, and post the admission decisions again.
8. All validations pertaining to changing the admission outcome status from PENDING to any valid admission outcome status are performed by the Automatic Batch Admission Decisions Import process.

   Examples of valid admission outcome statuses include the following:

   - CANCELLED
   - COND_OFFER
   - FUTURETERM
   - NO-QUOTA
   - OFFER
   - PENDING
   - REJECTED
   - VOIDED
   - WAITLIST
   - WITHDRAWN
9. If a validation fails, an error code is recorded in the interface table and the admission outcome status is not updated in the Oracle Student System target table.

Interface Tables

Oracle Student System uses the following interface tables during the Automatic Batch Admission Decisions Import process.

- IGS_AD_BATC_DEF_DET_ALL
- IGS_AD_ADMDE_INT_ALL

For information on table column descriptions and values, see Oracle Student System Table Information at http://metalink.oracle.com/metalink/plsql/ml2_gui.startup.

9

Offer Response Import Process

This chapter describes the process of importing admission offer responses into Oracle Student System tables. The following sections are in this chapter:

Definition

The Offer Response Import process allows institutions to import in bulk applicants' responses to admission offers from University and Colleges Admission Service (UCAS) to Oracle Student System production tables. This process also updates the offer response status, depending on each applicant's response (accepted, deferred, or rejected).

This process can also be used by institutions outside of the UK which do not use the Export Applications to OSS process.

The Export Applications to OSS process receives data from UCAS and populates the interface tables. The admission Offer Response Import process can then be run to import data from the interface tables into Oracle Student System tables.

The admission Offer Response Import process can also be run to import data from the interface table if required. Institutions can use their own methods to populate data in the interface tables.

Overview

UCAS transmits applicants' response to the institution's admission offer. The Export Applications to Oracle Student System process extracts applicant records from UCAS and places the data into an interface table. The interface table is designed for the purpose of transferring that data to Oracle Student System tables.

Oracle Student System provides an interface table into which UCAS admission offer responses are loaded. These offer response records are then imported into Oracle Student System tables and the offer response statuses in the Oracle Student System tables are updated. In addition, an exception report can be generated which includes all records which fail the import process.

To begin with, the Export Applications to OSS concurrent process downloads the admission offer responses from UCAS and places them in the interface table *IGS_AD_OFFRESP_INT*. It populates the *IGS_AD_OFFRESP_BATCH* table with details of the batch. It then submits a concurrent manager request - Import of Offer Response - which, in turn, performs a set of validations and then inserts the data into the *IGS_AD_PS_APPL_INST_ALL* table. This process also updates the admission offer response statuses.

When the import process returns an error, you can look at the error report and correct the records in the interface table, and then run the concurrent process to upload data into Oracle Student System.

The errors occurring during the import process a can be generated into a report and can be viewed by running the Import Details - Offer Response Details Error Report concurrent process.

References

For information on admission decisions and offers, see Chapter 4, Admissions, *Oracle Student System User's Guide.*

Admission Offer Response Import Process

The Admission Offer Response Import process proceeds as outlined below.

1. Users load admission offer response records from UCAS by running the Export Applications to OSS concurrent process into the interface table *IGS_AD_OFFRESP_INT* and populates the *IGS_AD_OFFRESP_BATCH* table.

   You can also populate the interface table and batch table using tools of your choice, if you are not using the Export Application to OSS concurrent process. While entering records into the interface tables, the status must be specified as 2 (Pending).

   For information on the concurrent process, see *Oracle Student System for the United Kingdom User's Guide.*

2. In addition to loading admission offer responses, the Export Applications to OSS process also downloads applications from UCAS and exports admission decisions to UCAS.

3. The Export Applications to Oracle Student System process then invokes the Import of Offer Response concurrent process.

4. The Import of Offer Response process looks for a batch identifier in the interface table, and starts the import process only when a valid batch identifier exists. The import process proceeds as described below.

5. The Import of Offer Response fetches all pending offer response records from the interface table.

6. The Import of Offer Response checks if the application corresponding to the offer response exists in the table *IGS_AD_PS_APPL_INST_ALL*. If not, the record is aborted, an error is logged, and the next record is taken up for import.

7. The following items are validated on the records being imported: Application Instance Outcome Status; Application Processing Status; Interface Offer Response Status; Offer Deferment Status; Actual Response Date; Authorized Date; Offers across admission process categories; Application Acceptance Condition; and Institution Code.

   The validations performed and the error messages logged are listed under Admission Offer Response Import Process Validations.

8. If the parameter "Do you want to import applications if the Offer Response deadline has passed?" is set to Yes, offer responses received after the deadline are imported, provided the response is ACCEPTED and it is received earlier than the current system date.

If the parameter is set to No, all offer responses received after the response deadline are rejected.

9. If any validation fails, an error is logged and the next record is processed, starting from step 5.
10. If the record passes all the validations, it is inserted into the production table.
11. The Authorized Date and Admission Offer Deferment Status columns in the table IGS_AD_PS_APPL_INST_ALL are updated as per the offer response.
12. If the actual response date is NULL, then the system date is taken as the actual response date.
13. When a record is successfully inserted, the preenrollment process is invoked with the confirmation indicator and eligibility indicator set to yes.
14. If preenrollment process is successful, then the record is deleted from the interface table and the record details are logged.

    Errors that could occur during the import process are listed in Admission Import Process Error Codes.

Interface Tables

Oracle Student System uses the following interface tables during the Admission Offer Response Import process.

- IGS_AD_OFFRESP_ERR
- IGS_AD_OFFRESP_INT
- IGS_AD_OFFRESP_BATCH

For information on table column descriptions and values, see Oracle Student System Table Information at http://metalink.oracle.com/metalink/plsql/ml2_gui.startup.

Admission Offer Response Import Process Validations

Table 9–1 lists the validations performed while importing offer response records. This information is useful for entering data into the IGS_PS_LGCY_UV_INT table.

Table 9–1 Validations performed while importing admission offer response records

Validation performed while importing offer responses	Error messages logged
PERSON_ID, ADMISSION_APPL_NUMBER, NOMINATED_COURSE_cD and SEQUENCE_NUMBER combination must exactly match one Admission Application instance.	Admission Application Instance does not exist
The offer response status populated in the Interface table must be associated with a valid System Offer Response Status	Offer Response Status is Invalid.
The Offer Response status in the Interface table is same as that of the Admission Application Instance	Offer Response Status has not changed
Offer response status cannot be updated from accepted to pending.	Offer Response cannot be updated from Accepted to Pending
Offer response status cannot be updated from accepted to rejected.	Offer Response cannot be updated from Accepted to Rejected
Offer response status cannot be updated from accepted to deferral.	Offer Response cannot be updated from Accepted to Deferral
Offer response status cannot be updated from rejected to accepted.	Offer Response cannot be updated from Rejected to Accepted
Offer response status cannot be updated from rejected to pending.	Offer Response cannot be updated from Rejected to Pending
Offer response status cannot be updated from rejected to deferral.	Offer Response cannot be updated from Rejected to Deferral
Offer deferment status is must be mapped to a system offer deferment status.	Offer Deferment Status is not mapped to a system status.
Actual response date must be later than the current system date.	Actual Response Date <= to the current date.
Actual response date must be specified.	Actual Response date cannot be null.
Intent to Attend Other Institution code must be an institution_cd from igs_or_institution.	Intent to Attend Other Institution code is Invalid.
If offer response status is changed from DEFERRAL to PENDING, then the deferred calendars must not be specified.	Deferred Academic Calendar and/or Sequence Number are Invalid
If offer response status is changed to DEFERRAL, then the deferred calendar and sequence number should both exist.	Deferred Academic Calendar or Sequence Number must exist.

Part III

Enrollments Module

10

Enrollment Registration Update Process

This chapter describes the enrollment registration update process. The following sections are in this chapter:

Definition

Institutions in the UK require a mechanism to record that a student has appeared for registration. The date of registration must be captured and if the student is pre-enrolled in any units, the unit attempts must be confirmed. The enrollment registration update process enables the institutions to update registration information and confirm pre-enrolled units.

Overview

Institutions create a list of students and their registration dates after gathering the information from registration forms. The file is uploaded to an interface table IGS_EN_REG_UPD_INT. It is the institution’s responsibility to upload data into the interface table using their own tools.

Institutions then run the Batch Registration Date Update concurrent process. The process takes the records from the interface table, and for each student, checks if there are any unconfirmed units under inactive programs, and changes their status from unconfirmed to enrolled. If any one unit attempt is confirmed the program attempt status is updated to ENROLLED.

This process returns an error if the student has no INACTIVE program attempts.

On successfully updating the unit attempts, the process updates the registration date as the enrollment date, and deletes the record from the interface table.

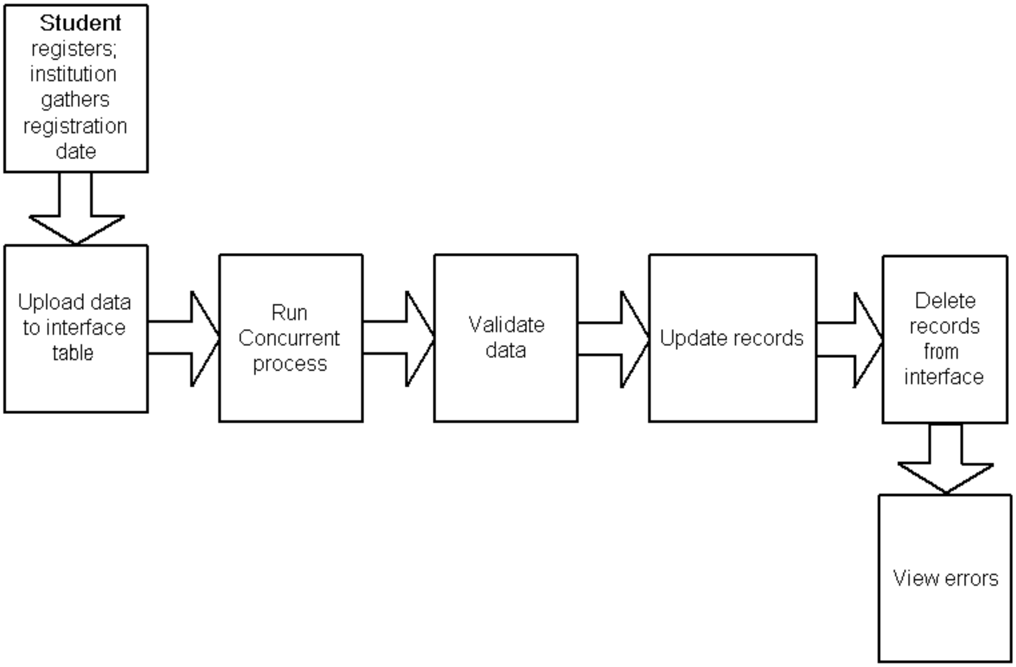
References

For information on enrollments, see *Oracle Student System for the United Kingdom User's Guide*.

Enrollment Registration Update Process Flowchart

Figure 10–1 shows the Enrollments Registration Update process from a functional perspective.

Figure 10–1 Enrollment Registration Update Process Flow


Student registers; institution gathers registration date
Upload data to interface table
Run Concurrent process
Validate data
Update records
Delete records from interface
View errors

Enrollment Registration Update Process

The enrollment registration update process generates an exception report either at the time of running the Batch Registration Date Update concurrent process (depending on the parameter setting) or as a separate concurrent process.

Prerequisites

The following constraints apply to the enrollment registration update process.

- All pre-enrollments must be carried out through the pre-enrollment concurrent process.
- The student registration details are uploaded into interface table IGS_EN_REG_UPD_INT.

Steps

The enrollment registration update process is outlined below.

1. Institutions enter data such as batch IDs into the IGS_EN_REG_BTCH_INT table. The batch IDs specified here must be used in the IGS_EN_REG_UPD_INT table which contains data to be imported.
2. Institutions enter data into the IGS_EN_REG_UPD_INT interface table provided by Oracle Student System. The institutions must devise their own means of entering data into the interface tables.

   This includes details such as person number, program code for the student's unit attempt, date of receiving enrollment form, and so on.

   Note: When no program details are passed in, the process tries to enroll the unit attempts associated with all programs. When specific program details are passed in, the process tries to confirm the unit attempts only for those programs passed in.
3. Institutions run the Batch Registration Date Update concurrent process by specifying the batch ID and enrollment method type.

   For information on this concurrent process and parameters, see *Oracle Student System for the United Kingdom User's Guide.*
4. All records which are not uploaded into OSS table are listed in an exception report at the end of the process.

Interface Tables

Student System uses the following interface tables during the enrollment registration update process.

- IGS_EN_REG_BTCH_INT
- IGS_EN_REG_UPD_INT

For information on table column descriptions and values, see Oracle Student System Table Information at
http://metalink.oracle.com/metalink/plsql/ml2_gui.startup.

11

Interactive Voice Response Process

This chapter describes the processes that support the interactive voice response (IVR) function. This chapter includes the following sections:

Definition

The APIs described in this chapter allow institutions that have interactive voice response (IVR) systems to communicate with Oracle Student System and perform enrollment functions. Oracle Student System provides these APIs which function as an interface between the institution's IVR and Oracle Student System database.

Overview

The IVR systems provided by the institutions send data such as the student's login to Oracle Student System and retrieve information which needs to be given to the student. This is carried out with the help of the APIs described in this chapter.

For the APIs to be able to communicate with Oracle Student System, IVR data attributes must be mapped to Oracle Student System data attributes. For instance, Oracle Student System API expects a call number, which maps to the IVR's unit code: IVR must pass the value of the unit code to call number and pass it back as the return parameter. APIs can only pass and accept Oracle Student System data and not IVR Codes.

Oracle Student System provides business views for the institution to map Oracle Student System data to IVR data , as described in the section Business Views.

The APIs provided support a range of functions including:

- validation routines for input data including person details, career / program of student, term Calendar, and person steps
- validating login
- list schedule
- viewing enrollment cart
- adding / removing unit from cart
- dropping a unit
- dropping all units
- enrolling into units
- clean up the cart
- update enrollment statistics

Institutions must set up the term calendars to be used in IVR. Institutions select from the list of term calendars in the Enrollment Setup window (Navigation: Enrollment - Setup - Enrollment Setup).

Institutions must also specify an enrollment method type to be used in IVR in the Enrollment Method Type window. (Navigation: Enrollment - Setup - Enrollment Method Type).

For information on how to set up enrollment and enrollment method type details, refer to *Oracle Student System User's Guide*.

Business Views

Oracle Student System requires the following data points for the IVR-enabled enrollment function:

- person number
- program type / career
- program code
- term calendar
- call number

- grading schema code
- enrolled credit points
- drop reasons

Where IVR uses different attribute names for these data points, they need to be mapped to the above using the business views provided. Users may also obtain additional data like these views, to provide to the student.

Table 11–1 Business views provided by Oracle Student System

Business View	Description
Person Information	Allows institutions to look up additional information about the student and map the IVR user to an FND user.
Career/Program Mapping	Allows the institution to look up information about the programs.
Calendar Period	Allows the institution to look up information about the calendar periods
Unit Section Occurrence	Allows institutions to look up instructor and occurrence information for a given unit section.
Unit Grading Schema; Unit Section Grading Schema	Allows institutions to look up the grading schemas for a unit section or, in case it is not set up at the section level, at the unit level.
Student Unit Attempt	Allows institutions to look up a student's unit attempt information.
IVR data Unit Section Version - Credit Points	Allows institutions to look up unit section level credit points limit, or, if that is not set up, unit level credit points limit.
Higher Education Unit Section	Allows institutions to look up additional information about a unit section and to map the unit section to the call number.
Student Program Attempt	Describes a student's attempt at studying a particular course offered by the university.
Discontinuation Status	Describes user-defined administrative unit statuses that map to system-defined unit statuses. Used to list the administrative unit status of the user when the drop function is performed.
Discontinuation Reason	Describes the reason that the user gives when dropping a section.

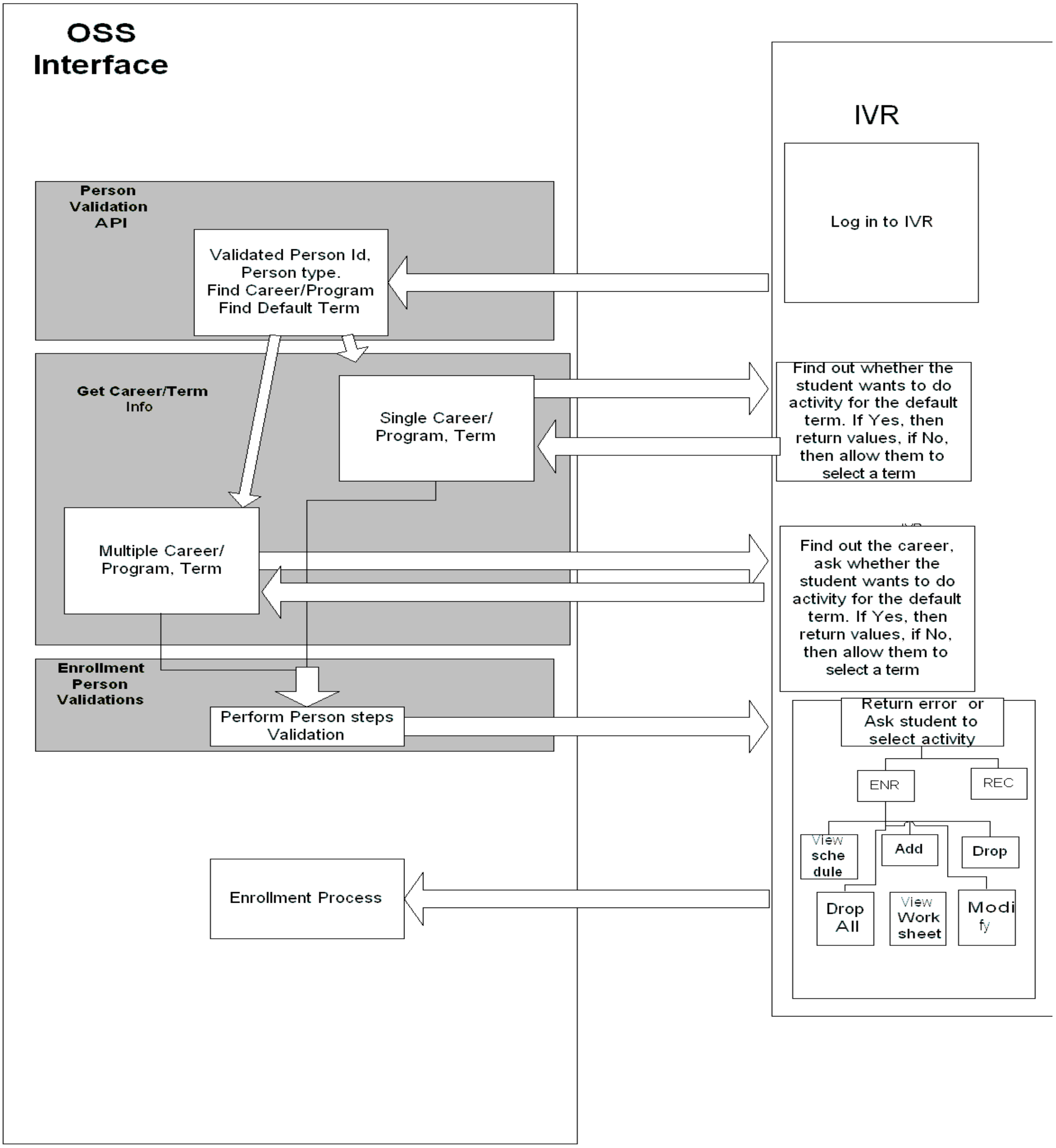
References

For information on enrollments, see Managing Enrollments, *Oracle Student System User’s Guide*.

IVR Process Functional Flowchart

Figure 11–1, page 11-12 shows the Oracle Student System - IVR interface provided by the APIs.

Figure 11–1 Interactive Voice Response Process Functional Flowchart


OSS Interface
Person Validation API
Validated Person Id, Person type. Find Career/Program Find Default Term
Get Career/Term Info
Single Career/ Program, Term
Multiple Career/ Program, Term
Enrollment Person Validations
Perform Person steps Validation
Enrollment Process
IVR
Log in to IVR
Find out whether the student wants to do activity for the default term. If Yes, then return values, if No, then allow them to select a term
Find out the career, ask whether the student wants to do activity for the default term. If Yes, then return values, if No, then allow them to select a term
Return error or Ask student to select activity
ENR
REC
View sche dule
Add
Drop
Drop All
View Work sheet
Modi fy

Calling the IVR Enrollment APIs

Oracle Student System provides APIs for the IVR system to call in order to perform enrollment functions. Institutions decide where and when the APIs will be called, although some of the APIs must be called before others in order for the validations to work.

The following sections describing various APIs are arranged in the order in which the APIs may be called. This section provides one possible flow for the institutions to follow, and outlines all of the APIs and how they work.

Institutions can look up the messages and determine the message type being passed by the API to the IVR system using the business view.

Prerequisites

From the person number, Oracle Student System API identifies whether the student has a single career/program or multiple careers/programs. The API would also default to a term for which the student can attempt enrollment.

1. Institutions must set up enrollment validations in Oracle Student System.
2. Institution must set up the enrollment method.
3. Institution must set up calendars.

Login API

Institution's IVR verifies the login information that the student provides, and passes the Person Number to the Oracle Student System Login API. The Login API returns a value Y indicating that the user can proceed or N indicating there is a login error, depending on the validation outcome.

The Login API takes the Person Number as input and outputs transaction status, number of messages, and encoded message text.

Get Career/Term Information API

From the person number, Oracle Student System API identifies whether the student has a single career/program or multiple careers/programs. The API would also default to a term for which the student can attempt enrollment.

If the default term is not what the student wishes to enroll for, it is up to the institution to set up IVR to allow users to select the term. Institutions can use the Enrollment setup window under Enrollments - Setup.

The Get Career/Term Information API requires Person Number as input. as input. It outputs program, default term, transaction status, number of messages, and encoded message text.

The Get Career/Term Information API must be called after the login API.

Enrollment Person Level Validation API

This API performs a person level validation based on the enrollment setup to determine whether the student is eligible to perform enrollment related tasks, such as adding or dropping units.

The Enrollment Person Level API requires Person Number, Program Code, and Alternative Code (Term) as input and outputs the status, number of messages, and encoded message text.

List Schedule API

This API retrieves and allows students to view enrollment schedule based on the career/program and the term selected.

This API requires Person Number, Program Code, and Alternate Code (Term) as input parameters, and outputs the transaction status, number of messages, encoded message text, and Call Number value set, which includes call number, credit point, grading schema, unit attempt status, and waitlist position.

The List Schedule API must be called after Login API, Get Program/Term Information API, and Enrollment Person Level Validation API.

Drop API

This API performs validation checks on requests to drop unit in a selected term for a career/program and drops the unit(s).

This API requires as input Person Number, Program Code, Alternative Code (Term), Call Number, and Drop Reason Code. It outputs transaction status, number of messages, and encoded message text.

The Drop API must be called after Login API, Get Program/Term Information API, and Enrollment Person Level Validation API.

Drop All API

This API performs the validation checks on requests to drop all units for a selected career / program and term.

This API requires Person Number, Program Code, Alternate Code (Term), and Drop Reason Code as input. It outputs transaction status, number of messages, and encoded message text.

The Drop All API must be called after the Login API, Get Program / Term Information API, and Enrollment Person Level Validation API.

List Enrollment Cart API

This API allows the IVR system to retrieve all unconfirmed units for the selected career / program and term.

This API requires Person Number, Program Code, and Alternate Code (Term) as input. It outputs transaction status, number of messages, and encoded message text.

The List Enrollment Cart API must be called after the Login API, Get Program / Term Information API, and Enrollment Person Level Validation API.

Purge Enrollment Cart API

The Purge Enrollment Cart API allows the IVR system to delete all unit sections that have been added to the enrollment cart through IVR.

This API requires Person Number, Program Code, and Alternate Code (Term) as input. It outputs transaction status, number of messages, and encoded message text.

The Purge Enrollment Cart API must be called after the Login API and Get Program / Term Information API.

Add API

This API allows the IVR system to help students add units for the term and program / career selected.

This API requires Person Number, Program Code, Alternate Code (Term), Call Number, and Audit Indicator as input. It outputs Waitlist Required Indicator, transaction status, number of messages, and encoded message text.

The waitlist required indicator indicates that the unit the student is adding is full and needs to be waitlisted if student wishes to proceed.

The Add API must be called after the Login API, Get Program / Term Information API, and Enrollment Person Level Validation API.

Waitlist API

This API allows the IVR system to put students on waitlist for units for the term and program / career selected. The student is later moved up from the waitlist to enrolled as per the auto-enroll process. For information, see Chapter 5, Enrollments, *Oracle Student System User's Guide*.

This API requires Person Number, Program Code, Alternate Code (Term), and Call Number as input. It outputs transaction status, number of messages, and encoded message text.

This API must be called after the Login API, Get Program / Term Information API, Enrollment Person Level Validation API, and Add API.

No Waitlist Count API

This API allows the IVR system to keep track of the number of students who reject being waitlisted.

This API requires Person Number, Call Number, Alternate Code for Term Calendar, and either Career or Program Code as input. It outputs transaction status, number of messages, and encoded message text.

This API must be called after the Login API, Get Program / Term Information API, Enrollment Person Level Validation API, and Add API.

Remove API

This API allows the IVR system to help students remove an unconfirmed unit for the term and program / career selected.

This API requires Person Number, Program Code, Alternate Code (Term), and Call Number as input. It outputs transaction status, number of messages, and encoded message text.

This API must be called after the Login API, Get Program / Term Information API, and Enrollment Person Level Validation API.

Enroll API

This API allows the IVR system to help students enroll for all units in the enrollment cart for the career / program and term selected. Units that do not need to be enrolled must be removed from the enrollment cart before calling this API.

This API requires Person Number, Program Code, and Alternate Code (Term) as input. It outputs transaction status, number of messages, and encoded message text.

This API must be called after the Login API, Get Program / Term Information API, and Enrollment Person Level Validation API.

Part IV

Financial Aid Module

12

Student Employment Payroll Data Upload Process

This chapter describes the process of uploading work-study-related payroll data from the Student Employment Payment Details Interface table to the Student Employment Payment Details target table in the Financial Aid subsystem of Oracle Student System. The following sections are in this chapter:

Definition

The Upload Payment Details process is registered as a concurrent program, which enables users to upload payroll data from the Student Employment Payment Details Interface table to the Student Employment Payment Details target table in the Financial Aid subsystem of Oracle Student System for tracking student employee earnings related to student work-study awards.

Overview

The Financial Aid subsystem of Oracle Student System administers student employment activities related to helping students secure employment, where employment supplements grants, loans, and scholarships in a student's financial aid package. When student employment involves Federal Work Study (FWS), strict Federal regulations and procedures govern the collection, disbursement, and tracking of Federal funds.

Work-study is designed to encourage part-time employment of undergraduate and graduate students who need income to help pay for the cost of education and training. The Federal government pays a majority of the students' salaries with the remainder provided by the employer, usually the institution. Work-study positions are often in campus offices and laboratories or combined with internships or research assistantships.

On many campuses, the responsibility for administration of student employment is shared between the Financial Aid Office and an employment or Career Services Office. Additionally, at many institutions, the Human Resources Office or payroll system tracks employee details.

For each work-study fund, a notification limit is defined, either as a percentage of the fund amount or as a dollar amount. The Upload Payment Details process initiates a workflow process to inform the student and the employer when student compensation approaches, reaches, or exceeds the maximum eligibility for the work-study fund in a particular period as defined in the fund setup.

The Oracle Student System Financial Aid subsystem offers an interface, which automates the sharing of data, between an external or internal legacy human resources / payroll system and the Oracle Student System Financial Aid subsystem in the Financial Aid Office.

The Student Employment Payroll Data Upload process begins when users transfer payroll data from an external or internal human resources / payroll system to the IGF_SE_PAYMENT_INT interface table. Then users run the Upload Payment Details concurrent process to transfer the payroll data from the IGF_SE_PAYMENT_INT interface table to the IGF_SE_PAYMENT Oracle Student System target table in Financial Aid.

The Upload Payment Details concurrent process is an inbound process only, in that payment data flows into the Financial Aid subsystem from an internal or external human resources / payroll system.

Payroll data imported into the Financial Aid subsystem is used to monitor whether work-study earnings have exceeded the allocation. If the paid amount exceeds the allocation, the payroll upload is rejected.

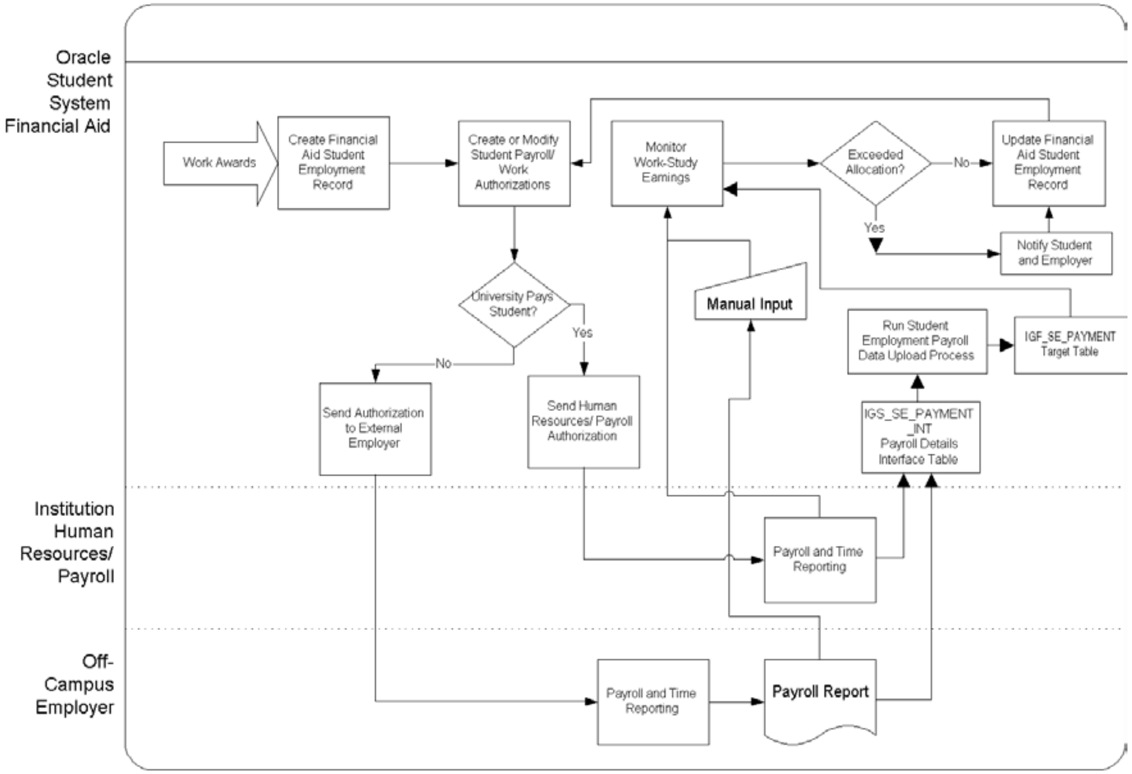
Reference

For additional information on the Student Employment Payroll Data Upload process, see Chapter 7, Financial Aid, *Oracle Student System User's Guide*.

Student Employment Work-Study Process Flowchart

Figure 12–1, page 12-4 shows the high-level student employment work-study process as described in the accompanying text.

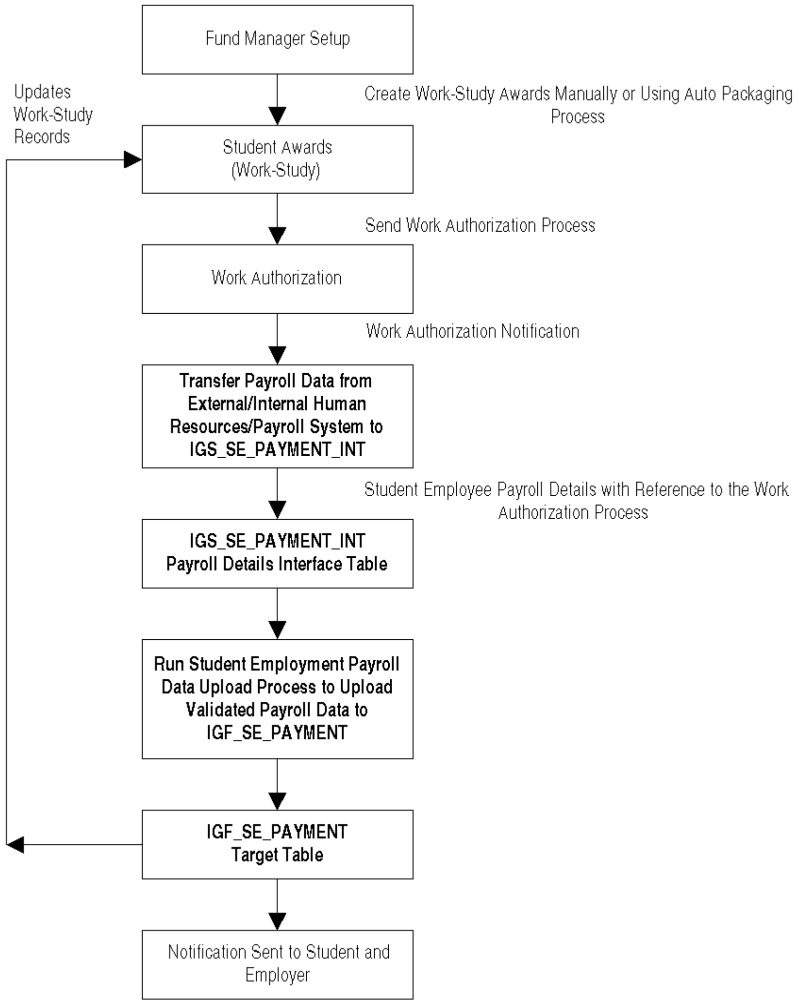
Figure 12–1 HIgh-Level Student Employment Work-Study Process Flowchart


Oracle Student System Financial Aid
Institution Human Resources/ Payroll
Off-Campus Employer
Work Awards
Create Financial Aid Student Employment Record
Create or Modify Student Payroll/ Work Authorizations
Monitor Work-Study Earnings
Exceeded Allocation?
No
Yes
Update Financial Aid Student Employment Record
Notify Student and Employer
University Pays Student?
Yes
No
Manual Input
Run Student Employment Payroll Data Upload Process
IGF_SE_PAYMENT Target Table
IGS_SE_PAYMENT _INT Payroll Details Interface Table
Send Authorization to External Employer
Send Human Resources/ Payroll Authorization
Payroll and Time Reporting
Payroll and Time Reporting
Payroll Report

Student Employment Payroll Data Upload Process Flowchart

Figure 12–2, page 12-5 shows the Student Employment Payroll Data Upload process as described in the Student Employment Payroll Data Upload Process section.

Figure 12–2 Student Employment Payroll Data Upload Process Flowchart


Fund Manager Setup
Create Work-Study Awards Manually or Using Auto Packaging Process
Updates Work-Study Records
Student Awards (Work-Study)
Send Work Authorization Process
Work Authorization
Work Authorization Notification
Transfer Payroll Data from External/Internal Human Resources/Payroll System to IGS_SE_PAYMENT_INT
Student Employee Payroll Details with Reference to the Work Authorization Process
IGS_SE_PAYMENT_INT Payroll Details Interface Table
Run Student Employment Payroll Data Upload Process to Upload Validated Payroll Data to IGF_SE_PAYMENT
IGF_SE_PAYMENT Target Table
Notification Sent to Student and Employer

Student Employment Payroll Data Upload Process

The Student Employment Payroll Data Upload process comprises the following steps:

1. Users transfer payroll data from an external or internal human resources/payroll system to the IGF_SE_PAYMENT_INT interface table.
2. Users then run the Upload Payment Details process to validate data.
3. Validated payroll data is uploaded from the IGF_SE_PAYMENT_INT interface table to the IGF_SE_PAYMENT Oracle Student System target table in Financial Aid. Oracle Student System adjusts the payment amounts against work-study funds.
4. If a payment is over the limit as established for the fund during setup, Oracle Student System sends notification to student and employer.
5. If failures occur during the Student Employment Payroll Data Upload process, the record is updated in the IGF_SE_PAYMENT_INT interface table with the relevant error code.

   Errors can be viewed in the error log.

Interface Table

Oracle Student System uses the following interface table during the Upload Payment Details process.

- IGF_SE_PAYMENT_INT

For information on table column descriptions and values, see Oracle Student System Table Information at http://metalink.oracle.com/metalink/plsql/ml2_gui.startup.

Part V

Student Finance Module

13

External Charges Import Process

This chapter describes the process of collecting and transferring external charges from the External Charges Interface table to the Charges target table in the Student Finance module of Oracle Student System. The following sections are in this chapter:

Definition

The External Charges Import process is a concurrent program that enables users to collect and transfer external charges from the External Charges Interface table to the Charges target table in the Student Finance module.

Overview

When billing a student, the Student Finance Office includes external charges, such as parking fees that originate outside the Student Finance Office and that have already been calculated by an external organization. The Student Finance Office uses third-party software or an external program to populate external charges for a particular student into the External Charges Interface table.

Users then run the Import External Charges Process that collects and transfers external charges from the External Charges Interface table to the Charges target table.

Prerequisites

To ensure that the Import External Charges process is able to correctly transfer external charges from the External Charges Interface Table, IGS_FI_EXT_INT_ALL, to the Charges target table, the following must be set up:

❏ Fee Type Calendar Instances

Fee type calendar instances are setup in the Fee Type Calendar Instances window.

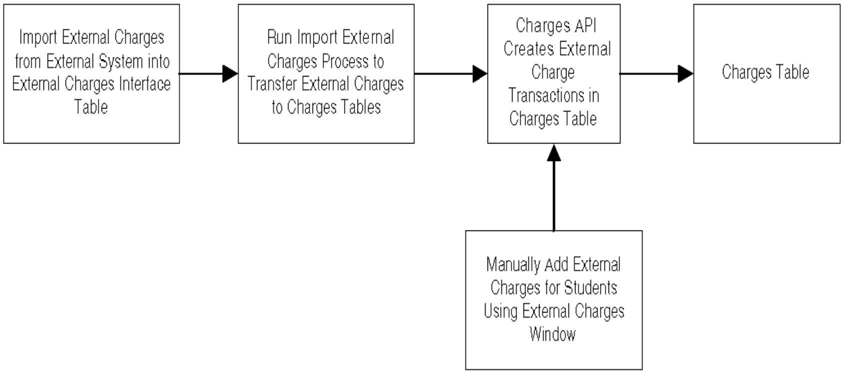
❏ Person Records

For information on creating person records and entering person details, see Chapter 3, Person, *Oracle Student System User's Guide*.

External Charges Import Process Flowchart

Figure 13–1, page 13-3 shows the External Charges Import process as described in the External Charges Import Process section.

Figure 13–1 External Charges Import Process Flowchart


Import External Charges from External System into External Charges Interface Table
Run Import External Charges Process to Transfer External Charges to Charges Tables
Charges API Creates External Charge Transactions in Charges Table
Charges Table
Manually Add External Charges for Students Using External Charges Window

External Charges Import Process

The External Charges Import process comprises the following steps:

1. Users run third-party software or an external program to populate external charges into the External Charges Interface Table, IGS_FI_EXT_INT_ALL.
2. Users run the Import External Charges process to collect and transfer external charges from the External Charges Interface Table, IGS_FI_EXT_INT_ALL, to the Charges target table.

   Note: Only records with the status TODO are imported.
3. The Import External Charges process logs the following types of errors:
   - Person is not an Active Student
   - Fee Type is Invalid
   - Fee Type Calendar Instances are Invalid
   - Currency code in the interface table does not match the value setup in System Options
   - GL Date is invalid
   - Flexfield columns are invalid.
   - Invalid value for the override_dr_ccid (code combination ID).
   - Invalid value for the override_cr_ccid (code combination ID).

4. Upon validating the records, the process calls the Charges API, which creates records in the Charges target table.
5. Users can add new external charges for a student directly into the Charges target table, thereby bypassing the Import External Charges process.
6. To add external charges for a student without running the Import External Charges process, users can enter external charges manually using the External Charges window.
7. In Oracle Student System, navigate to the External Charges window as follows:

   Student Finance - External Charges

   - The External Charges window has a descriptive flexfield available for user customization.
8. External charges that are imported by the Import External Charges process and those that are manually entered using the External Charges window are subsequently transferred to the student account by the Charges API.
9. The status of records which are successfully imported into the Charges table is updated to “Success”.

Interface Table

Oracle Student System uses the following interface table during the External Charges Import process.

- IGS_FI_EXT_INT_ALL

For information on table column descriptions and values, see Oracle Student System Table Information at http://metalink.oracle.com/metalink/plsql/ml2_gui.startup.

14

Import and Calculate Ancillary Charges Process

This chapter describes the process of collecting ancillary attributes for a student from the Ancillary Attributes Interface table, calculating the ancillary charges for those attributes using a rate table, and transferring those calculated charges to the Import Charges target table in the Student Finance module of Oracle Student System. The following sections are in this chapter:

Definition

The Import and Calculate Ancillary Charges process is a concurrent program that enables users to collect ancillary attributes from the Ancillary Attributes Interface table, calculate the ancillary charges for those attributes, and transfer the calculated charges to the Import Charges target table in the Student Finance module.

Overview

When billing a student, the Student Finance Office includes ancillary charges, such as dining and housing fees. The dining and housing attributes of a student are provided by an office external to the Student Finance Office. The Student Finance

Office uses an external program to populate the ancillary attributes of the student into the Ancillary Attributes Interface table.

Users then run the Import and Calculate Ancillary Charges concurrent program that collects ancillary attributes for a student from the Ancillary Attributes Interface table, calculates the ancillary charges for those attributes using a rate table, and transfers the calculated charges from the Ancillary Attributes Interface table to the Import Charges target table. The Charges API is called upon successful import of ancillary charges, and an ancillary charge records are created in the charges table.

Prerequisites

To ensure that the Import and Calculate Ancillary Charges process is able to correctly calculate and transfer ancillary charges, the following must be set up:

❏ Person Records

For information on creating person records and entering person details, see Creating Person Details Procedure, Person Detail Procedures, *Oracle Student System User's Guide*.

❏ Local currency must be set up in the system Options form.

❏ An Active Fee Type of System fee type "ANCILLARY" must be set up.

❏ Fee Types Operating in a Specific Fee Period Known as Fee Type Calendar Instances.

The fee type calendar instances defined are used to create ancillary charges. Fee type calendar instances are defined in the Fee Type Calendar Instances window.

To define fee type calendar instances, see Entering Data Applicable to a Fee Type in a Calendar Instance, Fee Setup, *Oracle Student System User's Guide.*

❏ Accounting Information (Receivable Accounting Flex Field and Revenue segments) must be defined at FTCI level or at the System Options level.

❏ Ancillary Segments for Fee Type Calendar Instances

The segments defined are used as dynamic prompts for defining the rates for the combination of ancillary attributes in the Rate table. Ancillary segments are set up in the Ancillary Segments window.

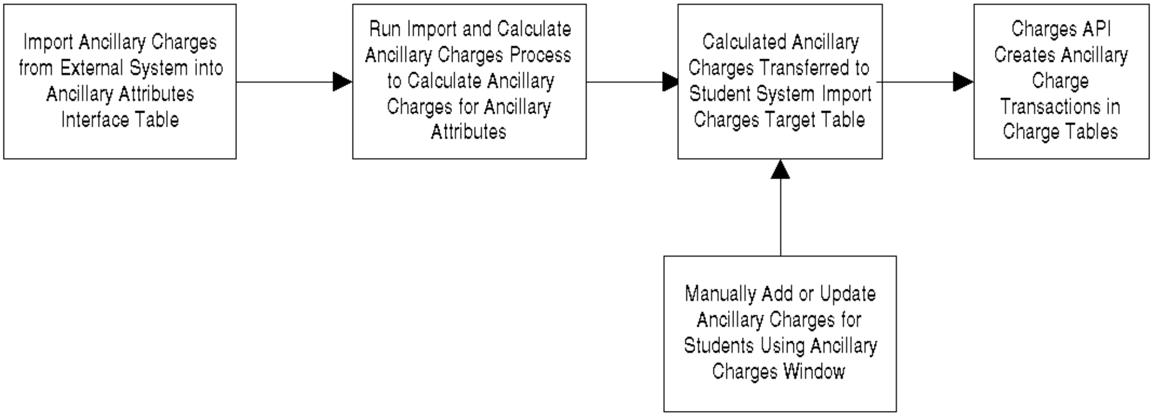
❏ Ancillary Rates for the Different Combinations of Attributes Defined in the Ancillary Segments Window

The ancillary rates defined are used by the Ancillary Charge Import & Calculation process to calculate the ancillary charges for a particular student. Ancillary rates are set up in the Ancillary Rates window.

Import and Calculate Ancillary Charges Process Flowchart

Figure 14–1, page 14-3 describes the Import and Calculate Ancillary Charges process as described in the Import and Calculate Ancillary Charges Process section.

Figure 14–1 Import and Calculate Ancillary Charges Process Flowchart


Import Ancillary Charges from External System into Ancillary Attributes Interface Table
Run Import and Calculate Ancillary Charges Process to Calculate Ancillary Charges for Ancillary Attributes
Calculated Ancillary Charges Transferred to Student System Import Charges Target Table
Charges API Creates Ancillary Charge Transactions in Charge Tables
Manually Add or Update Ancillary Charges for Students Using Ancillary Charges Window

Import and Calculate Ancillary Charges Process

The Import and Calculate Ancillary Charges process comprises the following steps:

1. Users run third party software or an external program to populate the ancillary attributes into the Ancillary Attributes Interface Table, IGS_FI_ANC_INT_ALL.
2. Users run the Import and Calculate Ancillary Charges concurrent program to calculate the ancillary charges based on the ancillary rates setup and transfer them from the Ancillary Attributes Interface table, IGS_FI_ANC_INT_ALL, to the Import Charges target table.
3. The Import and Calculate Ancillary Charges process logs the following types of errors:

- No Setup of Ancillary Rates for the Combination of Ancillary Attributes Imported
- Invalid Data Items Imported
- No Active Fee Type of System fee type "ANCILLARY" exists
- No Active FTCI exists for the Ancillary Fee type.
- In active person is provided to the process.
- Person Id Type is provided and Alternate person Id is not provided.
- Person Id Type is not provided and Alternate person Id is provided.
- Person Number, Person Id Type and Alternate person Id are provided.
- DFF Definition is invalid or Flex field Columns are invalid.

These errors can be rectified using the Ancillary Charges Error Resolution window.

4. To rectify errors, in Oracle Student System, navigate to the Ancillary Charges Error Resolution window as follows:

   Student Finance - Ancillary Charges Error Resolution

   For additional information on resolving errors using the Ancillary Charges Error Resolution window, see Chapter 6, Student Finance, *Oracle Student System User's Guide*.

5. Users can add new details about a student directly into the Import Charges table, thereby bypassing the Import and Calculate Ancillary Charges process.

   To add ancillary attributes for a student without running the Import and Calculate Ancillary Charges Process, users can enter ancillary charges in the Ancillary Charges window. The ancillary charges are then calculated and transferred to the Import Charges target table.

6. To enter ancillary charges, in Oracle Student System, navigate to the Ancillary Charges window as follows:

   Student Finance - Ancillary Charges

   For additional information on adding ancillary attributes using the Ancillary Charges window, see Chapter 6, *Oracle Student System User's Guide*.

7. Ancillary charges that are calculated by the Import and Calculate Ancillary Charges process and those that are manually entered in the Ancillary Charges

window are subsequently transferred to the student account upon successfully completing the import process.

Interface Table

Oracle Student System uses the following interface table during the Import and Calculate Ancillary Charges process.

- IGS_FI_ANC_INT_ALL

For information on table column descriptions and values, see Oracle Student System Table Information at http://metalink.oracle.com/metalink/plsql/ml2_gui.startup.

15

Credits API

This chapter describes the specifications and methodology for using the Credits API. The Credits API provides an extension to the existing functionality of creating credit transactions using windows and processes in the Student Finance module of Oracle Student System. The following sections are in this chapter:

Definition

The Credits API is a public API that can be used internally or externally to create payment or credit information in the Student Finance module of Oracle Student System.

Overview

The Credits API enables users to do the following:

- create credit transactions
- import payments or credit information from an external legacy system

The Credits API is available for use by all subsystems in Oracle Student System, legacy applications, or third-party applications and can be accessed as standard PL/SQL server-side routine calls.

The Credits API is responsible for updating Holds Balance whenever a credit transaction is created for a person. Hence, it cannot be invoked simultaneously with the Holds Balance Conversion Process.

Users invoke the Credits API for each credit record transaction to be created and transferred to the Credits tables. The Credits API checks the validity of all parameters, resulting in a return status, which informs users of the results of the API operation.

Credits API Usage

Users call the IGS_FI_CREDITS_API_PUB.Create_Credit routine to create credit transactions. Credit transactions in Student Finance include payments, sponsorship credits, lockbox receipts from Oracle Receivables, and financial aid credits.

The Credits API returns credit_id and credit_activity_id of the credit transaction created as out parameters. Table 15–1, page 15-2 shows the type and number of parameters in the Credits API.

Table 15–1 Type and Number of Parameters in the Credits API

Type of Parameter	Number of Parameters
Standard API Input Parameters	4
Credit Input Parameters	28 + 1 (descriptive flexfield parameter)
Standard API Output Parameters	3
Credit Output Parameters	2

Validation Rules

Table 15–2, page 15-3 specifies the validation rules for complex Credits API parameters.

Table 15–2 Validation Rules for Complex API Parameters

Parameters	Validation Rule
party_id	The Party ID must be a valid party in the HZ_PARTIES table.
credit _type_id	The Credit Type ID must be a valid Credit Type ID in the IGS_FI_CR_TYPES_ALL table. The effective date of the credit transaction must lie between the credit type start date and end date, if not null. If the Credit Class value associated with the Credit Type is Internal Financial Aid or External Financial Aid, the credit transaction being created is a Financial Aid credit and the validation rule is enforced.
duplicate records	If the combination of party_id and credit number on this credit transaction matches any existing credit transactions which have not been reversed, the error message IGS_FI_CREDIT_DUPLICATE needs to be raised. This check also exists in the table level unique constraint.

Prerequisites

Before calling the Credits API, users must set up the following entities in the Student Finance module:

❏ Credit Type

❏ Desired Lookup Values for Credit Instrument and Credit Source Parameters

❏ Valid currency: FND_CURRENCIES_ACTIVE_V

❏ Lockbox: AR_LOCKBOXES

❏ Invoice ID: IGS_FI_INV_INT

❏ Active award category calendar instance

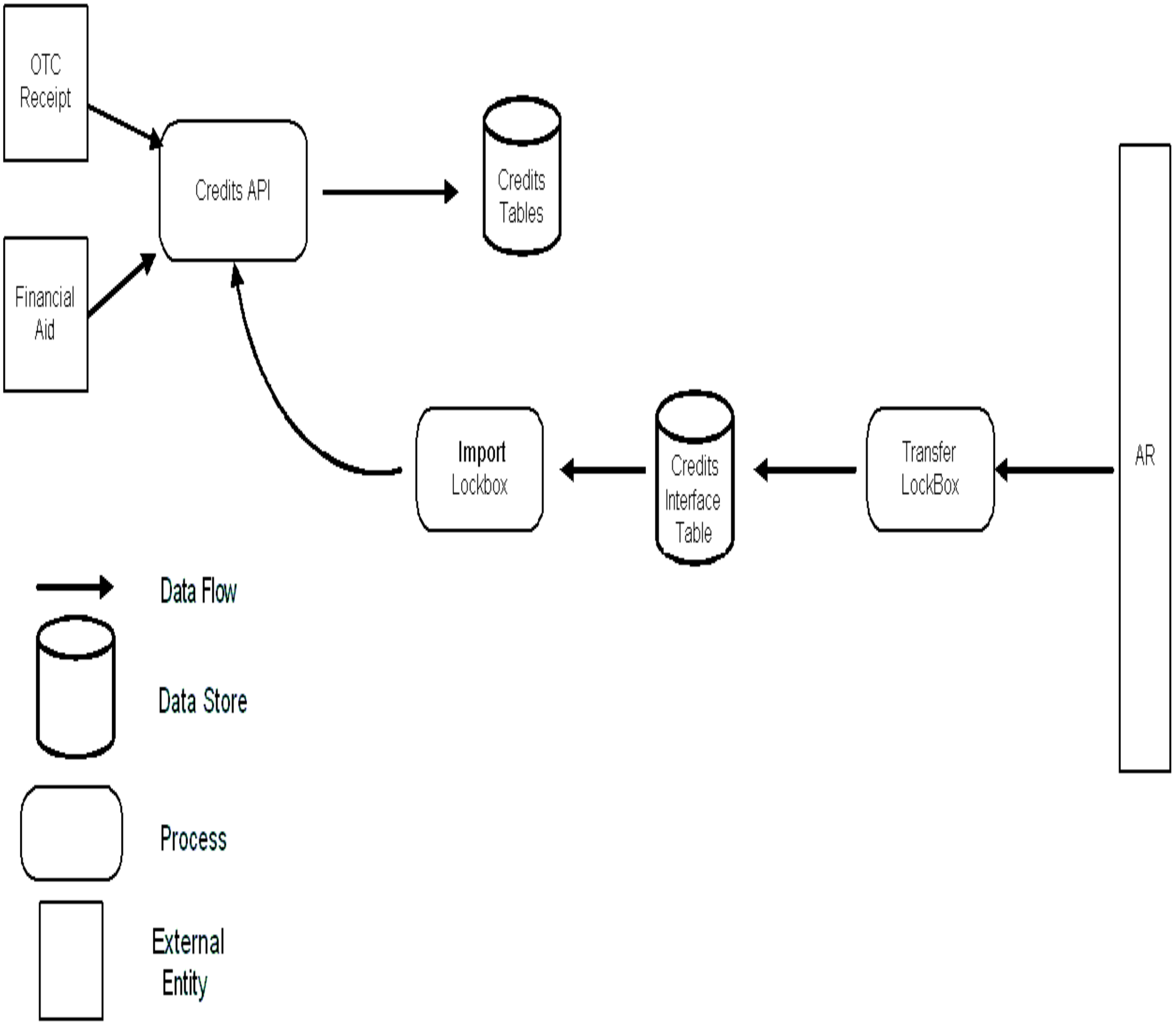
❏ Active fee calendar instance

❏ GL Date: open, future date

Credits API Process Flowchart

Figure 15–1, page 15-4 shows the Credits API process flowchart as described in the Overview and Credits API sections of this chapter.

Figure 15–1 Credits API Process Flowchart



Credits API Processing Logic

The following is a high-level description of the Credits API processing logic.

1. To create credit transactions, invoke the Credits API.
2. Validate the data, item level and row level, according to the validation rules specified in the Validation Rules section of this chapter.
3. Insert valid records into the IGS_FI_CREDITS_ALL table.
4. If the currency code is not local, translate the currency into local currency by multiplying the amount and exchange rate.
5. The Credits API calls the Build Account process, which is a PL/SQL package called IGS_FI_PRC_ACCT_PKG.Build_Accounts, that generates debit and credit account pairs for credit transactions.
6. Insert valid record into the IGS_FI_CR_ACTIVITES table.
7. The Standard Balance and Holds Balance amount is updated for the credit amount for the credit transaction created by the Credits API.

Return Status Messages

The Credits API's return status informs users of the results of the API operation. The return status values are as follows:

- success; FND_API.G_RET_STS_SUCCESS
- error; FND_API.G_RET_STS_ERROR
- unexpected error; FND_API.G_RET_STS_UNEXP_ERROR

An error return status indicates that the Credits API failed to perform some or all of the operations requested by the caller. An error return status is usually accompanied by messages describing how to fix the errors.

An unexpected error return status indicates that the Credits API has encountered an error condition it did not expect or could not handle and subsequently discontinued processing. Examples of unexpected errors include the following:

- irrecoverable data inconsistency errors
- memory errors
- programming errors

Credits API Parameters

Table 15–3, page 15-6 lists the Credits API parameters.

Table 15–3 Credits API Parameters

Parameter	Type	Data Type	Required	Description
p_api_version	IN	NUMBER	Yes	In the current version of the API, users must pass a value of 1.0 for this parameter.
p_init_msg_list	IN	VARCHAR2		allows API callers to request that API does initialization of the message list on their behalf
p_commit	IN	VARCHAR2		used by API callers to ask the API to commit on their behalf
p_credit_number	IN	VARCHAR2	Yes	credit number for the credit to be created Default: none Validate: credit number should not be null Destination: IGS_FI_CREDITS_ALL. CREDIT_NUMBER
p_credit_status	IN	VARCHAR2		status of the credit transaction Default: CLEARED Validate: only valid value is CLEARED Destination: IGS_FI_CREDITS_ALL.STATUS, IGS_FI_CR_ACTIVITIES.STATUS

Table 15–3 Credits API Parameters

Parameter	Type	Data Type	Required	Description
p_credit_source	IN	VARCHAR2	Yes	credit source of the transaction Default: none Validate: validate with lookup codes from lookup type IGS_FI_CREDIT_SOURCE Destination: IGS_FI_CREDITS_ALL. CREDIT_SOURCE
p_party_id	IN	NUMBER	Yes	party (person) ID for the credit transaction Default: none Destination: IGS_FI_CREDITS_ALL.PARTY_ID
p_credit_type_id	IN	NUMBER	Yes	identifies the credit type of the credit transaction Default: none Validate: validates the existence of this credit type as valid Destination: IGS_FI_CREDITS_ALL. CREDIT_TYPE_ID
p_credit_instrument	IN	VARCHAR2	Yes	identifies the credit instrument of the credit Default: none Validate: validate with lookup codes from lookup type IGS_FI_CREDIT_INSTRUMENT Destination: IGS_FI_CREDITS_ALL. CREDIT_INSTRUMEMNT

Table 15–3 Credits API Parameters

Parameter	Type	Data Type	Required	Description
p_description	IN	VARCHAR2		description of the credit transaction Default: NULL Validate: none Destination: IGS_FI_CREDITS_ALL. DESCRIPTION
p_amount	IN	NUMBER	Yes	credit transaction amount Default: none Validate: >0 IGS_FI_CAPI_CRD_AMT_NULL Destination: IGS_FI_CREDITS_ALL. AMOUNT, IGS_FI_CR_ACTIVITIES .AMOUNT
p_currency_cd	IN	VARCHAR2		local currency of the credit transaction Default: local currency Validate: needs to be a valid currency Destination: IGS_FI_CREDITS_ALL. CURRENCY_CD
p_exchange_rate	IN	NUMBER	Required if currency is not local.	exchange rate for the currency other than local Default: if the currency is local, then default is 1 Validate: none Destination: IGS_FI_CREDITS_ALL.EXCHANGE_RATE after translation into local currency

Table 15–3 Credits API Parameters

Parameter	Type	Data Type	Required	Description
p_receipt_lockbox_number	IN	VARCHAR2		receipt LockBox number for all credits created from AR LockBox Transaction process Default: none Validate: should be a valid LockBox number in AR_LOCKBOXES table with STATUS = A Destination: IGS_FI_CREDITS_ALL.RECEIPT_LOCKBOX_NUMBER
p_transaction_date	IN	DATE		transaction date for the entered credit transaction Default: system date Validate: none Destination: IGS_FI_CREDITS_ALL.TRANSACTION_DATE, IGS_FI_CR_ACTIVITIES.TRANSACTION_DATE
p_effective_date	IN	DATE		effective date for this credit transaction Default: transaction date Validate: none Destination: IGS_FI_CREDITS_ALL.EFFECTIVE_DATE

Table 15–3 Credits API Parameters

Parameter	Type	Data Type	Required	Description
p_source_transaction_id	IN	NUMBER		tracks the source transaction; used to track the imported AR LockBox transactions Default: NULL Validate: none Destination: IGS_FI_CREDITS_ALL.SOURCE_TRANSACTION_ID
p_credit_card_code	IN	VARCHAR2		credit card type; VISA, MasterCard Default: NULL Validate: validate with lookup codes from lookup type IGS_CREDIT_CARDS Destination: IGS_FI_CREDITS_ALL. CREDIT_CARD_CODE
p_credit_card_holder_name	IN	VARCHAR2		name of the credit card holder Default: NULL Validate: none Destination: IGS_FI_CREDITS_ALL. CREDIT_CARD_HOLDER_NAME
p_credit_card_number	IN	VARCHAR2		credit card number for credit card transaction Default: NULL Validate: none Destination: IGS_FI_CREDITS_ALL. CREDIT_CARD_NUMBER

Table 15–3 Credits API Parameters

Parameter	Type	Data Type	Required	Description
p_credit_card_expiration_date	IN	DATE		credit card expiration date Default: NULL Validate: > or = system date Destination: IGS_FI_CREDITS_ALL. CREDIT_CARD_EXPIRATION_DATE
p_credit_card_approval_code	IN	VARCHAR2		credit card approval code as obtained from any verifying agency Default: NULL Validate: none Destination: IGS_FI_CREDITS_ALL. CREDIT_CARD_APPROVAL_CODE
p_attribute_record	IN	attribute_rec_type		descriptive flexfield record Default: NULL Validate: none Destination: IGS_FI_CREDITS_ALL. ATTRIBUTE1 TO 20
p_credit_id	OUT	NUMBER	Yes	credit ID of the credit transaction created by the API call
p_credit_activity_id	OUT	NUMBER	Yes	credit activity ID for the credit transaction created
x_return_status	OUT	VARCHAR2		represents the API overall return status
x_msg_count	OUT	NUMBER		number of messages in the API message list

Table 15–3 Credits API Parameters

Parameter	Type	Data Type	Required	Description
x_msg_data	OUT	VARCHAR2		message in encoded format if x_msg_count = 1
p_invoice_id	IN	NUMBER		invoice identifier for the charge transaction for which the negative adjustment credit transaction is created
p_awd_yr_cal_type	IN	VARCHAR2		award year calendar for the financial aid credits
p_awd_yr_ci_sequence_number	IN	NUMBER		award year calendar instance identifier
p_fee_cal_type	IN	VARCHAR2		fee calendar for the financial aid credits
p_fee_ci_sequence_number	IN	NUMBER		fee calendar instance identifier
p_v_check_number	IN	VARCHAR2		This parameter can be provided irrespective of the Credit Class of the transaction and when the credit instrument is Check
p_v_source_tran_type	IN	VARCHAR		This parameter is required for a deposit transaction with ‘Enrollment Deposit’ credit class.
p_v_source_tran_ref_number	IN	NUMBER		This parameter is required for a deposit transaction with ‘Enrollment Deposit’ credit class.

Table 15–3 Credits API Parameters

Parameter	Type	Data Type	Required	Description
p_d_gl_date	IN	DATE		This parameter is required for self-service pages calling the credits API. The GL date passed must be the current system date, and it would overwrite any existing value.

16

Student Refund Transfer Process

This chapter describes the outbound process of transferring student refund data between the Student Finance module of Oracle Student System and Payables. The following sections are in this chapter:

Definition

The Student Refund Transfer process includes the Transfer Refunds to Interface process which collects and transfers student refunds from the Refunds table in the Student Finance module to the Refunds Interface target table.

Student refunds are excess funds after payment of charges. Excess credits on students' accounts are refunds that are returned to students after charges are paid.

Overview

The Student Refund Transfer process described in this chapter is applicable if Oracle Payables is not configured to deal with Oracle Student System refunds. If Oracle Payables is configured, you can run the Transfer Refunds to Oracle Payables process which transfers the refunds records directly to Oracle Payables Invoice interface tables.

For information on Oracle Payables integration with student finance, see Chapter 6, Student Finance, *Oracle Student System User's Guide*.

The process described in this chapter is to be followed when GL is not installed, and the refunds are processed by third party payables software.

The Student Refund Transfer process begins by running the Transfer Refunds Information to Refunds Interface process to collect and transfer the refund data from the Refunds table to the Refunds Interface target table. Running the Transfer Refunds Information to Refunds Interface process creates refund information in the Refunds Interface table. The Refunds Interface table stores refund transaction information that interacts with third party check generation system or an external Payables system for processing refunds externally.

System Options

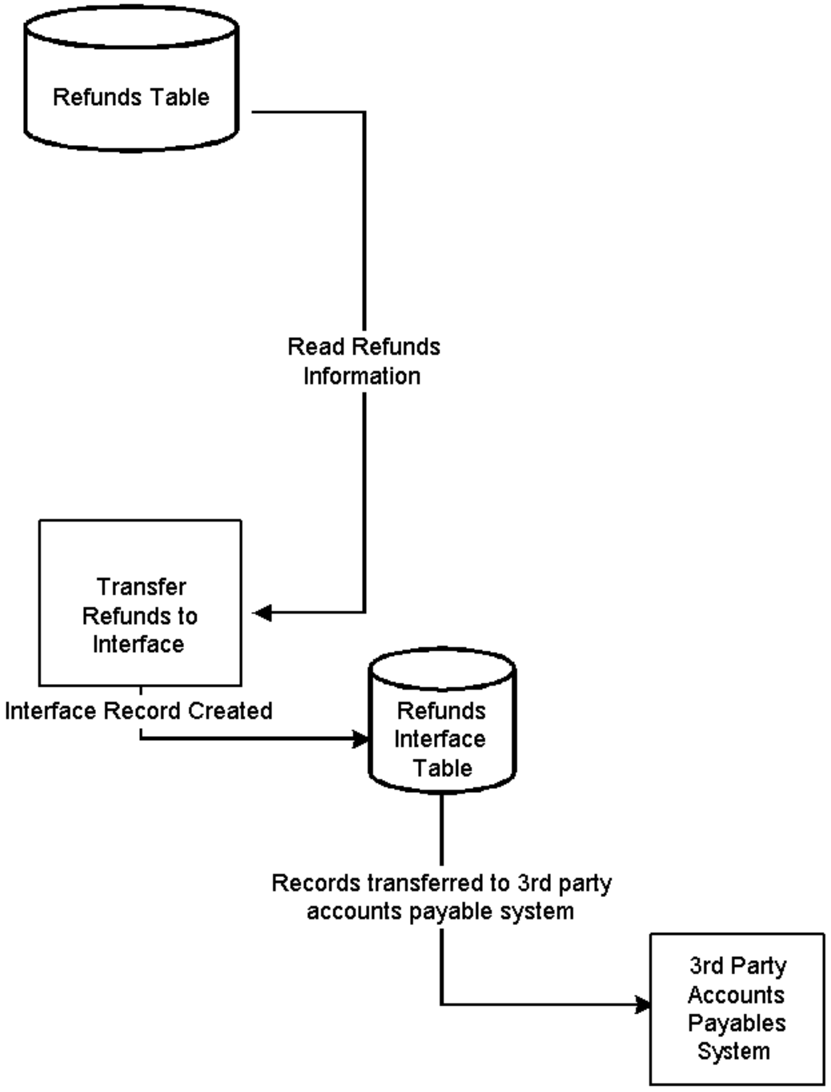
The following system option must be specified with regard to the refunds function:

- In the System Options window, Refunds tab, specify Other as the Refunds Destination.

  For more information on system options, refer to Chapter 6, Student Finance, *Oracle Student System User's Guide*.

Student Refund Transfer Process Flowchart

Figure 16–1 Student Refund Transfer Process Flowchart


Refunds Table
Read Refunds Information
Transfer Refunds to Interface
Interface Record Created
Refunds Interface Table
Records transferred to 3rd party accounts payable system
3rd Party Accounts Payables System

Student Refund Transfer Process

The Process Refunds of Excess Credits concurrent process must be run before the Student Refund Transfer process.

The Student Refund Transfer process includes the following steps:

1. Users run the Transfer Refunds Information to Refunds Interface process to create refund information in the Refunds Interface table (outbound). Running this process collects and transfers refund data from the Refunds table to the Refunds Interface target table, IGS_FI_REFUNDS_INT_ALL.
2. Users employ third party tools to transfer student refund data from the Refunds Interface target table to Payables for payment processing.
3. Oracle Student System recommends that users refrain from creating new records in IGS_FI_REFUND_INT_ALL.

Interface Table

Oracle Student System uses the following interface table during the Student Refund Transfer process.

- IGS_FI_REFUNDS_INT_ALL

For information on table column descriptions and values, see Oracle Student System Table Information at http://metalink.oracle.com/metalink/plsql/ml2_gui.startup.

17

Posting Interface Process

This chapter describes the process of transferring student finance transactions from three source tables in the Student Finance module of Oracle Student System to the Posting Interface target table. The following sections are in this chapter:

Definition

The Posting Interface process is a concurrent program that enables users to transfer transactions with accounting implications from three areas in the Student Finance module of Oracle Student System to the Posting Interface target table in the Student Finance module.

Overview

Users run the Posting Interface concurrent program, which is a batch process, to collect and transfer all eligible journal entry transactions for posting purposes from the Charges, Credits, and Application source tables to the Posting Interface target table.

The Posting Interface process is an outbound process only, in that journal entry transactions ultimately flow out from the Student Finance module to another application or other financial systems.

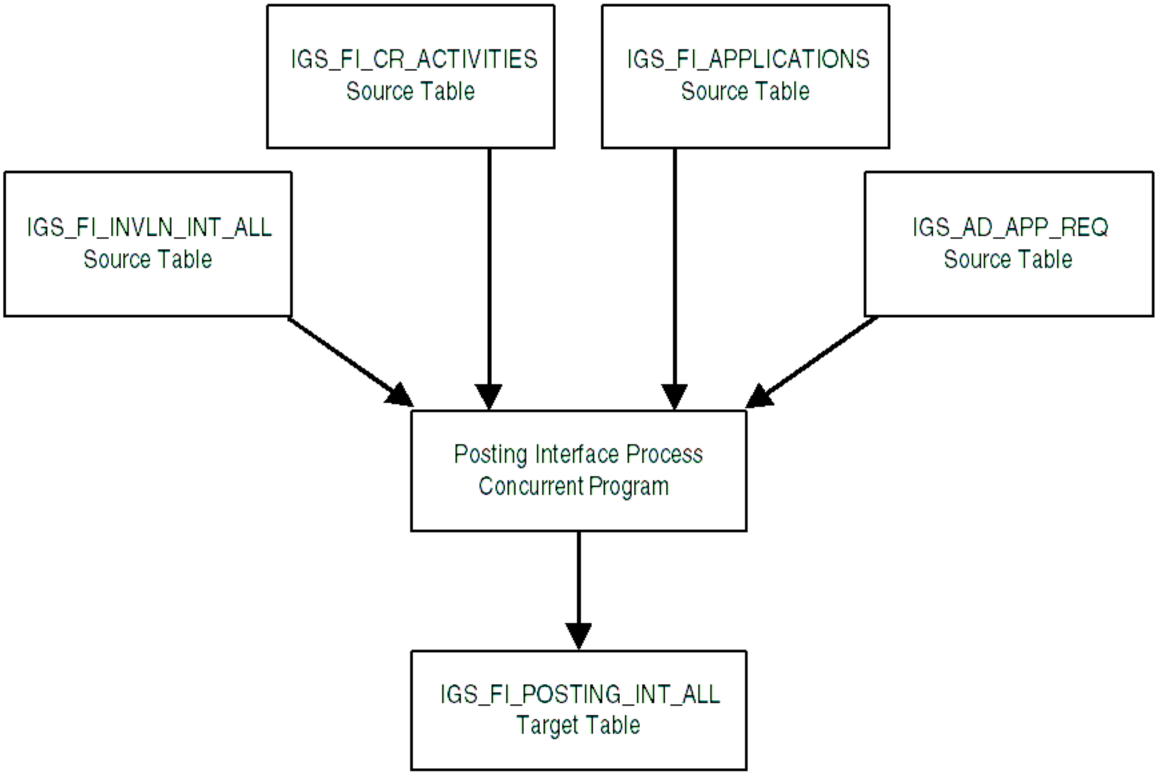
Note: Posting Interface is available to users who do not have Oracle Financials installed.

The Posting Interface target table in Student Finance is not an open interface table. As such, users cannot insert, delete, or modify records in the Posting Interface table.

Posting Interface Process Tables Flowchart

Figure 17–1, page 17-3 shows the interaction of the source tables and the Posting Interface table as described in the Processing Logic section.

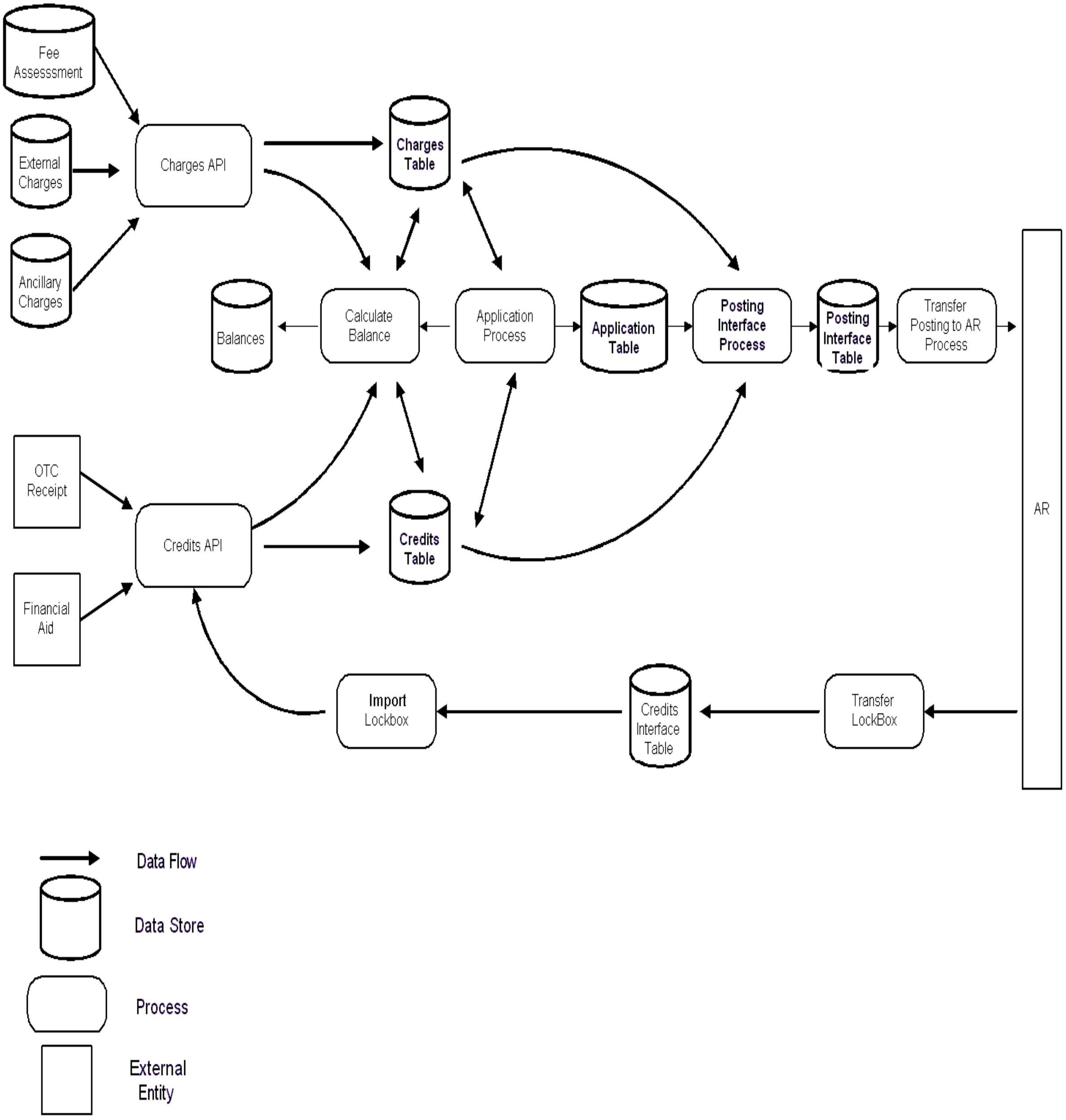
Figure 17–1 Posting Interface Process Tables Flowchart



Posting Interface Process Flowchart

Figure 17–2, page 17-4 describes the Posting Interface process flowchart as described in the accompanying text.

Figure 17–2 Posting Interface Process Flowchart



Processing Logic

The following is a high-level description of the processing logic of the Posting Interface process.

1. If users have selected the Accrual accounting method during setup, journal entry transactions are selected from the following tables when the Posting Interface concurrent program is run:
   - IGS_FI_INVLN_INT_ALL
   - IGS_FI_CR_ACTIVITIES
   - IGS_FI_APPLICATIONS
   - IGS_AD_APP_REQ
2. If users have selected the Cash accounting method during setup, journal entry transactions are selected from the following tables when the Posting Interface concurrent program is run:
   - IGS_FI_CR_ACTIVITIES
   - IGS_FI_APPLICATIONS
   - IGS_AD_APP_REQ
3. The Posting Interface process collects data from the appropriate source tables and transfers it to the Posting Interface target table based on specified concurrent program parameters.
4. The Posting Interface process collects debit and credit account pairs in either accounting flexfield format or alphanumeric account code strings.
5. Journal entry transaction data is created in the IGS_FI_POSTING_INT _ALL target table with the mapping information indicated in Table 17–1.
6. All newly created records in the IGS_FI_POSTING_INT _ALL target table have a status value of TODO. When users run the Transfer Posting to AR process, the process collects and transfers records from the IGS_FI_POSTING_INT _ALL target table to the AR Interface table and the value of the Status field changes from TODO to Transferred.
7. The process log for the Posting Interface process logs the following information:
   - transaction number
   - transaction date
   - debit and credit transaction amount

- debit and credit accounts
- transaction type

Mapping of Source and Target Columns

Table 17–1 Mapping of Source and Target Columns

Source Table/Concurrent Process Parameter	Column in Source Table/Data Value	Target Table	Column in Target Table
Concurrent Parameter		IGS_FI_POSTING_INT_ALL	BATCH_NAME
Concurrent Parameter		IGS_FI_POSTING_INT_ALL	POSTING_CONTROL_ID
Concurrent Parameter	GL_DATE	IGS_FI_POSTING_INT_ALL	ACCOUNTING_DATE
Respective Tables	TRANSACTION_DATE, EFFECTIVE_DATE, APPLY_DATE, FEE_DATE	IGS_FI_POSTING_INT_ALL	TRANSACTION_DATE
Respective Tables	CURRENCY_CD	IGS_FI_POSTING_INT_ALL	CURRENCY_CD
Respective Tables	DR_ACCOUNT_CD	IGS_FI_POSTING_INT_ALL	DR_ACCOUNT_CD
Respective Tables	CR_ACCOUNT_CD	IGS_FI_POSTING_INT_ALL	CR_ACCOUNT_CD
Respective Tables	AMOUNT	IGS_FI_POSTING_INT_ALL	AMOUNT
	CREDIT, CHARGE, APPLICATION, APPLFEE	IGS_FI_POSTING_INT_ALL	SOURCE_TRANSACTION_TYPE
Respective Tables IGS_FI_INVLN_INT, IGS_FI_CR_ACTIVITIES, IGS_FI_APPLICATIONS	INVOICE_LINES_ID, CREDIT_ACTIVITY_ID, APPLICATION_ID	IGS_FI_POSTING_INT_ALL	SOURCE_TRANSACTION_ID
	‘TODO’	IGS_FI_POSTING_INT_ALL	STATUS

Interface Table

Oracle Student System uses the following interface target table during the Posting Interface process.

- IGS_FI_POSTING_INT_ALL

For information on table column descriptions and values, see Oracle Student System Table Information at http://metalink.oracle.com/metalink/plsql/ml2_gui.startup.

18

Sponsorship Assignment API

This chapter describes the Sponsorship Assignment API. This API provides a means to create multiple student-sponsor assignment records without manual entry. The following sections are in this chapter:

Definition

The Sponsorship Assignment API is a public API that can be used to assign a number of students to a sponsor automatically. Oracle Student System allows you to manually create student-sponsor assignment records though the interface. This API is a means to automatically create multiple student-sponsor assignment records.

Overview

The Sponsorship Assignment API can be accessed as standard PL/SQL server-side routine calls.

Users invoke the Sponsorship Assignment API for each student to be assigned to a sponsor. The Sponsorship Assignment API checks the validity of all parameters, resulting in a return status, which informs users of the result of the API operation.

The student must exist in the system and the sponsor must be set up. A financial aid base record must also exist for the student. However, if a base record does not exist, the Sponsorship Assignment API calls the Base Record - Create FA Base Record process to create a base record before creating the student-sponsor assignment record. The record does not include any Pays Only details.

The Sponsorship Assignment API can be executed in a loop to create multiple student-sponsor records.

Sponsorship Assignment API Usage

Users call the IGF_SP_ASSIGN_PUB.CREATE_STDNT_SPNSR_REL routine to create student-sponsor assignment records. Users can execute this in a loop to create multiple records.

The following parameters must be specified:

- Either person ID or alternate person ID type and alternate ID
- Sponsor code
- Award year and award year sequence
- Load calendar and load calendar sequence
- Amount of sponsorship

The Sponsorship Assignment API returns the status of the function call: S if the record is successfully created, E if there is an error, and U if there is an unexpected error.

Prerequisites

Before calling the Sponsorship Assignment API, users must ensure that:

❏ Sponsor setup with the award year and load calendars is completed.

❏ Student exists as a valid person in the system.

❏ For UK-specific use of this API, users must associate the system-defined person ID type SLC (student loan company) to a user-defined person ID type. Users must also define alternate person ID details for each person being assigned to a sponsor.

❏ User-defined person ID type must be unique.

Sponsorship Assignment API Processing Logic

The following is a high-level description of the Sponsorship Assignment API processing logic.

1. To create student-sponsor assignment records, invoke the Sponsorship Assignment API.
2. Validate the data, item level and row level, according to the validation rules specified under Validation Rules.
3. Create financial aid base record where a record doesn't exist for the specified student.
4. Create student-sponsor record with the amount specified as a parameter in the IGF_SP_SPDNT_REL_ALL table.
5. Return completion status to the calling process in order to go to the next record.

Sponsorship Assignment API Parameters

Table 18–1, page 18-11 lists the Sponsorship Assignment API parameters.

Table 18–1 Sponsorship Assignment API Parameters

Parameter	Type	Data Type	Required	Description
P_API_VERSION	IN	NUMBER	Yes	In the current version of the API, users must pass a value of 1.0 for this parameter.
P_INIT_MSG_LIST	IN	VARCHAR2	DEFAULT FND_API.G_FALSE	Allows API callers to request for message stack initialization.
P_COMMIT	IN	VARCHAR2	DEFAULT FND_API.G_FALSE	Standard Public API Parameter to check if the current transactions have to be committed explicitly.
X_RETURN_STATUS	OUT	VARCHAR2 (1)	Yes	Standard public API parameter to convey public API return status.

Table 18–1 Sponsorship Assignment API Parameters

Parameter	Type	Data Type	Required	Description
X_MSG_COUNT	OUT	NUMBER	Yes	Standard public API parameter contains message count returned by the public API.
X_MSG_DATA	OUT	VARCHAR2 (2000)	Yes	Standard public API parameter which contains the messages in the encoded format. Users must decode the error messages when they wish to see the actual message text.
P_PERSON_ID	IN	NUMBER	Conditionally mandatory	Person ID of the student being assigned to the sponsor. Either P_PERSON_ID or both P_ALT_PERSON_ID_TYPE and P_API_PERSON_ID must be specified.
P_ALT_PERSON_ID_TYPE	IN	VARCHAR2	Conditionally mandatory	User-defined person ID type. For UK-specific SLC requirements, user-defined value mapped to system-defined person ID type of SLC needs to be passed as this parameter. Either P_PERSON_ID or both P_ALT_PERSON_ID_TYPE and P_API_PERSON_ID must be specified.

Table 18–1 Sponsorship Assignment API Parameters

Parameter	Type	Data Type	Required	Description
P_API_PERSON_ID	IN	VARCHAR2	Conditionally mandatory	Alternate ID parameter in combination with system-defined person ID type is used to identify a unique person. Either P_PERSON_ID or both P_ALT_PERSON_ID_TYPE and P_API_PERSON_ID must be specified.
P_SPONSOR_CODE	IN	VARCHAR2	Yes	Active sponsor code pre-defined in the system for an award year calendar instance.
P_AWD_CI_CAL_TYPE	IN	VARCHAR2	Yes	Award calendar type for which the sponsor is offering sponsorships to students.
P_AWD_CI_SEQUENCE_NUMBER	IN	NUMBER	Yes	Award calendar instance sequence number for which the sponsor is offering sponsorships to students.
P_LD_CAL_TYPE	IN	VARCHAR2	Yes	Load/term calendar type under award calendar instance for which the sponsor is offering sponsorships to students.
P_LD_CI_SEQUENCE_NUMBER	IN	NUMBER	Yes	Load/term calendar instance sequence number for which the sponsor is offering sponsorships to students.

Table 18–1 Sponsorship Assignment API Parameters

Parameter	Type	Data Type	Required	Description
P_AMOUNT	IN	NUMBER	Yes	Amount that the sponsor is willing to provide a student for the specified term / load calendar instance with in an award calendar instance.

Validation Rules

The following validations are performed on the parameters passed with the Sponsorship Assignment API:

- Person ID specified must exist in the system.
- Alternate ID type must have a unique attribute.
- Alternate ID must exist, and it must be within the active date range.
- Sponsor code must be specified, and it must not be closed.
- Award year type and award year sequence must be specified, and the combination of award year type and sequence must be valid.
- Sponsorship amount must be specified and it must be a non-negative number.
- Load calendar type and load calendar sequence must be specified, and the combination of load calendar type and sequence must be valid.
- If person ID is specified, then alternate ID type and alternate ID **must not** be specified. If person ID is not specified, then both alternate ID type and alternate ID must be specified.
- Sponsor code - award year combination must be valid according to the sponsor setup
- The combination of person, award year, and load calendar must not already exist for the specified sponsor.

Part VI

Program Structure and Planning Module

19

Scheduling Interface Process

This chapter describes the process of transferring Unit Section Occurrences (USO) data between the Oracle Student System database and third-party scheduling software using scheduling interface tables. The following sections are in this chapter:

- Definition
- Overview
- Scheduling Interface Process Flowchart
- Scheduling Interface Import Process
- Interface Tables
- Scheduling API Parameters

Definition

This chapter describes the process of transferring data between the Oracle Student System database and third-party scheduling software. This process is usually completed by personnel appointed by the institution to schedule courses or Unit Section Occurrences (USOs).

Overview

The Scheduling Interface application enables data related to USOs to be passed between the Oracle Student System database and the third-party scheduling software to enable the scheduling software to take advantage of additional functionality such as class location, room, and building characteristic preferences. The scheduling software is used to review allocations and to make modifications before returning final room assignments to Oracle Student System through the interface table. The Oracle Student System scheduling interface enables the user to schedule, update, and cancel class room and building assignments through a third-party's scheduling software.

Table 19–1, page 19-2 describes the four components of the Oracle Student System Scheduling Interface.

Table 19–1 Oracle Student System Scheduling Interface Components

Component	Description
Oracle Student System database	Oracle Student System production tables
Oracle Student System concurrent processes	Enable communication between the Oracle Student System production tables and the Scheduling Interface tables. These processes pass data required by third-party scheduling software, and retrieve scheduling information into Oracle Student System tables through scheduling interface tables.
Scheduling APIs	Oracle provides an API which can be used by third-party scheduling software to update the scheduled class room and building for a Unit Section Occurrence. Scheduling software can get the data from the interface table by writing SQL statement or designing a custom API.
Third-party scheduling software	Scheduling software designed by a third-party used for assigning class room and building for USOs.

The components required to complete the scheduling cycle are as follows:

- Oracle Student System Concurrent Processes
- Scheduling API
- Third-Party Scheduling Software

Oracle Student System Concurrent Processes

The Export Setup Data for Scheduler concurrent process produces text files containing information about buildings, rooms, and locations that third-party scheduling software uses for the purpose of scheduling. The text file names are identified not by hard coded filenames but by lookup types and lookup codes. The lookup type is SCHED_EXP_FILES, and the lookup codes are IGS_PS_SCH_ROOM (data about rooms), IGS_PS_SCH_BLDG (data about buildings), and

IGS_PS_SCH_LOC (data about location). The concurrent request ID is appended to the meaning of the lookup codes.

The files containing the data are placed in a directory specified in the profile option “Path of the file where export data will be stored”.

The Initiate Scheduling of Unit Section Occurrences concurrent process passes USO data to scheduling interface tables from which the third-party scheduling software takes it and creates schedules. This process also passes on data related to various scheduling preferences set up by users from the Oracle Student System production tables into the new columns and tables added to the Scheduling Interface tables.

The following data may be passed on:

- Dedicated, Preferred, or Scheduled Building and Room
- Multiple Instructors
- Facilities
- To Be Announced (TBA) status
- USO Start Date and USO End Date
- Enrollment Expected and Override Max criteria attached to the unit section
- Cross Listed Unit Section
- Meet with Class Unit Section
- Unit Section Occurrence Reference

The Get Scheduled Interface Data takes records from the Scheduling Interface table and updates the Oracle Student System production tables with the USO class room and building assignments.

Oracle Student System can initiate REQUEST, UPDATE, and CANCEL transactions. When the transactions are successfully written to the interface table by the Oracle Student System API, the third-party scheduler can access the open interface table by devising an API or a SQL query. Oracle Student System provides an API that the scheduling software can use to update or insert records into the scheduling interface tables upon successfully scheduling class room and building for the USOs.

Scheduling API

A third-party scheduling API reads data from the Oracle Student System interface tables and passes it on to the third-party scheduling software. The third-party scheduling software processes the data and assigns a building and room for each scheduled USO, and effective time for to-be-announced USOs.

In addition to the data on USOs provided in the scheduling interface tables, the scheduling software also uses information on available buildings and rooms provided in flat files. Users may also enter the data on buildings and rooms into third-party software directly, without using the flat files.

Scheduling APIs provided by Oracle take information related to the class room and building returned by the third-party scheduling software, and passes it on to the Scheduling Interface tables and updates the Schedule Status of each record in the Scheduling Interface as OK if the third-party software successfully assigned a class room and building for the USO. If the third-party software is unsuccessful in assigning a class room and building to any USO, then the Scheduling API updates the Schedule Status of that specific record as ERROR.

The scheduling API includes two procedures:

- IGS_PS_SCH_INT_API_PUB.update_schedule, which updates existing records in the scheduling interface based on the data received from the third-party software.
- IGS_PS_SCH_INT_API_PUB.Insert_schedule, which inserts new records in the scheduling interface table based on the data received from the third-party software.

The parameters and the values to be used for calling these procedures are described under Scheduling API Parameters.

Third-Party Scheduling Software

Scheduling software designed by a third-party and used to assign class room and building for each USO.

Example

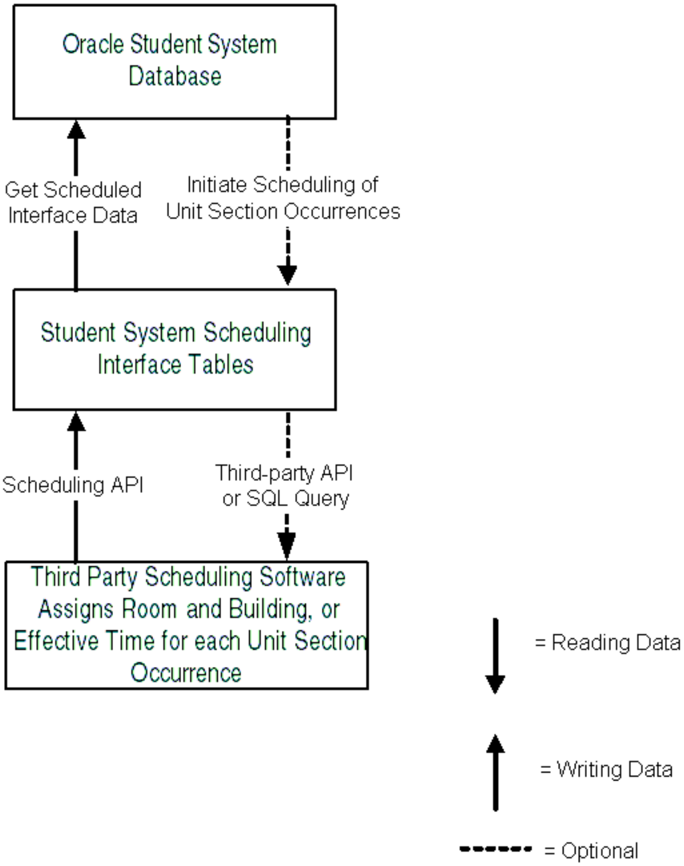
A Chemistry unit must have a specific laboratory in a particular building. The scheduler indicates that the laboratory is a Dedicated building and room for all USOs under the unit and unit section. However, a History or Mathematics unit may not require a specific room or building but may have a preference for the unit to be taught in a particular building or room. The scheduler may indicate that the requirement is a Preferred building or room for the USO. The scheduler can enter values manually to ensure that a specific room or building is scheduled.

The third-party scheduling software assigns a specific class room and building, or effective time for the USO and the Oracle Student System API updates the Oracle Student System production tables with the assignment. The buildings and rooms returned by the third-party scheduling software are treated as SCHEDULED buildings and rooms for the respective USOs.

Scheduling Interface Process Flowchart

Figure 19–1 shows the Scheduling Interface process flowchart as described in the accompanying text.

Figure 19–1 Scheduling Interface Process Flowchart


Oracle Student System Database
Get Scheduled Interface Data
Initiate Scheduling of Unit Section Occurrences
Student System Scheduling Interface Tables
Scheduling API
Third-party API or SQL Query
Third Party Scheduling Software Assigns Room and Building, or Effective Time for each Unit Section Occurrence
= Reading Data
= Writing Data
= Optional

Scheduling Interface Import Process

1. The Export Setup Data for Scheduler concurrent process can be run to provide setup information for the scheduling software in text files.

   Users may enter this data directly into the scheduling software if they choose to.

2. Data for scheduling meetings for USOs in the Scheduling Interface is stored in various placeholders in the Oracle Student System production tables.

3. To transfer data regarding USOs that require scheduling from the Oracle Student System database to the Scheduling Interface tables, the Initiate Scheduling of Unit Section Occurrences concurrent program must be run.

   - To run the program, users enter values for the Teaching Period and Schedule Type parameters. The list of values for the schedule type are Cancel the Scheduled Unit Section Occurrences, Reschedule Unit Section Occurrences, and Schedule Active and Unscheduled Unit Section Occurrences. When users choose any of these options, the concurrent process initiates transfer of the relevant data from the Oracle Student System database to the Scheduling Interface tables.

4. The third-party scheduling software extracts data from the interface tables and text files using custom APIs or SQL queries. The information is transferred to the third-party scheduling software.

   Users may initiate scheduling activity without taking data from the interface tables. Users may enter the data on USOs directly into third-party software.

5. The third-party scheduling software processes the data received from scheduling interface tables and text files, and assigns class room and building.

6. The Scheduling API provided by Oracle Student System collects the final class room and building assignments processed by the third-party scheduling software and moves the data into the Scheduling Interface tables. The Scheduling API updates the building ID, room ID, schedule status, and error text columns in the Scheduling Interface tables.

   Schedule status can be OK or ERROR. If the value is OK, then building ID, and room ID are mandatory for scheduled USOs.

   Start time, end time, building ID, and room ID can be null for USOs for which schedule is to be announced. Building ID and Room ID must be null for cancelled USOs.

   If the schedule status is ‘Error’, then ERROR_TEXT must be passed.

   The parameters of the Scheduling API are described in detail under Scheduling API Parameters.

7. Scheduling is now complete.

   - The Get Scheduled Interface Data concurrent program is run in Oracle Student System.
   - The process checks the Schedule Status field in the Scheduling Interface tables and updates the corresponding fields in the Oracle Student System

production tables with the class room and building assignments, schedule status, and error text. Once the Oracle Student System API has successfully transferred the data from the interface table to the production tables, the schedule status becomes SCHEDULED.

- If the Oracle Student System API finds null values in any of the fields, an error is returned.
- If Oracle Student System encounters any errors while saving the OK transactions in the Oracle Student System database, the Get Scheduled Interface Data concurrent process updates the schedule status as ERROR and updates the error text field with the error message returned by the system. For example, the error message could be, “The returned Building and Room have Null values”, “The returned Building and Room from third-party scheduling software do not exist in Oracle Student System”, or it could be the standard Oracle Applications/Database messages generated due to an application or database error. In the case of USOs with TBA (To Be Announced) status where the field’s Start Time, End Time, Building, and Room are intended to be null, the Oracle Student System API shows the schedule status as TBA and not as ERROR. The ERROR_TEXT message is “Building/Room not assigned by the Third Party Scheduling software as the USO has TBA status”.

8. Records that cannot be updated into the scheduling interface tables or into Oracle Student System tables can be taken up for a second cycle of processing using the same process described in Steps 4 and 5. The custom API designed by the third-party software must make provision for a second cycle of processing for dealing with records that fail to be updated the first time.

   - If a second cycle of processing takes place, Scheduling API collects the revised class room and building assignments from the third-party scheduling software and updates the schedule status of the corresponding records as OK or ERROR in the Scheduling Interface, as described in Step 6.

9. The Get Schedule Interface Data concurrent program can be run to read the OK and ERROR transactions from the Scheduling Interface tables. This process does not read transactions with a Schedule status of SCHEDULED, as scheduling has already completed successfully for those transactions and the schedule status is already updated as SCHEDULED by the process in the previous cycle.

   Get Scheduled Interface Data concurrent process, after reading the OK and ERROR transactions again from the Scheduling Interface tables, updates the Oracle Student System production tables with the class room and building assignments. It also updates the schedule status as SCHEDULED for successful updates, as described in Step 7. If the revised schedule status is SCHEDULED, the value previously stored in the ERROR_TEXT field becomes null.

The process described in steps 8, and 9 can optionally continue through user intervention until all the records in the Scheduling Interface show the schedule status as SCHEDULED, or until all the records in the Scheduling Interface are purged by the user.

10. Run the Purge Scheduling Interface Data concurrent process to remove data from the interface table. Currently you can only purge data from the past teaching calendar.

Interface Tables

Oracle Student System uses the following interface tables during the scheduling interface import process:

- IGS_PS_PREFS_SCH_INT_ALL
- IGS_PS_SCH_FACLT_ALL
- IGS_PS_SCH_HDR_INT_ALL
- IGS_PS_SCH_INSTR_ALL
- IGS_PS_SCH_INT_ALL
- IGS_PS_SCH_MWC_ALL
- IGS_PS_SCH_USEC_INT_ALL
- IGS_PS_SCH_X_USEC_INT_ALL

For information on table column descriptions and values, see Oracle Student System Table Information at http://metalink.oracle.com/metalink/plsql/ml2_gui.startup.

Scheduling API Parameters

Table 19–2 describes parameters and values to be used for calling the scheduling API update procedure, IGS_PS_SCH_INT_API_PUB.update_schedule. Table 19–3, page 19-11 describes parameters and values to be used for calling the scheduling API insert procedure, IGS_PS_SCH_INT_API_PUB.insert_schedule.

Table 19–2 Scheduling API update procedure parameters

Parameter Name	IN/ OUT	Data Type	Require d	
P_API_VERSION	IN	NUMBER	Yes	Standard API parameter
P_INIT_MSG_LI ST	IN	VARCHA R2	No	Standard API parameter
P_COMMIT	IN	VARCHA R2	No	Standard API parameter
P_VALIDATION _LEVEL	IN	NUMBER	No	Standard API parameter
P_INT_OCCURS _ID	IN	NUMBER	Yes	Unique identifier for a unit section occurrence in the interface table
P_START_TIME	IN	DATE	No	Scheduled start time of the Unit Section Occurrence
P_END_TIME	IN	DATE	No	Scheduled end time of the Unit Section Occurrence
P_BUILDING_ID	IN	NUMBER	No	Scheduled building identifier of the Unit Section Occurrence
P_ROOM_ID	IN	NUMBER	No	Scheduled room identifier of the Unit Section Occurrence
P_SCHEDULE_S TATUS	IN	VARCHA R2	Yes	Scheduling status of the Unit Section Occurrence.
P_ERROR_TEXT	IN	VARCHA R2	No	Error text to be passed when scheduling a Unit Section Occurrence fails
P_ORG_ID	IN	NUMBER	No	Operating unit identifier
X_RETURN_STA TUS	OUT	VARCHA R2	Yes	Standard API parameter
X_MSG_COUNT	OUT	NUMBER	Yes	Standard API parameter
X_MSG_DATA	OUT	VARCHA R2	Yes	Standard API parameter

Table 19–3 Scheduling API insert procedure parameters

Parameter Name	IN/ OUT	Data Type	Require d	
P_API_VERSION	IN	NUMBER	Yes	Standard API parameter
P_INIT_MSG_LI ST	IN	VARCHA R2	No	Standard API parameter
P_COMMIT	IN	VARCHA R2	No	Standard API parameter
P_VALIDATION _LEVEL	IN	NUMBER	No	Standard API parameter
P_TRANSACTIO N_TYPE	IN	VARCHA R2	Yes	Transaction type of the procedure call: REQUEST, COMPLETE, UPDATE, OR CANCEL
P_CAL_TYPE	IN	VARCHA R2	Yes	Calendar type of unit section.
P_SEQUENCE_N UMBER	IN	NUMBER	Yes	Calendar instance sequence number of the unit section
P_CAL_START_ DT	IN	DATE	Yes	Calendar start date of the unit section
P_CAL_END_DT	IN	DATE	Yes	Calendar end date unit section
P_UOO_ID	IN	NUMBER	Yes	Unit section identifier
P_INT_OCCURS _ID	IN	NUMBER	Yes	Unique identifier for a unit section occurrence in the interface table
P_START_TIME	IN	DATE	No	Scheduled start time of the Unit Section Occurrence
P_END_TIME	IN	DATE	No	Scheduled end time of the Unit Section Occurrence
P_BUILDING_ID	IN	NUMBER	No	Scheduled building identifier of the Unit Section Occurrence
P_ROOM_ID	IN	NUMBER	No	Scheduled room identifier of the Unit Section Occurrence
P_SCHEDULE_S TATUS	IN	VARCHA R2	Yes	Scheduling status of the Unit Section Occurrence.
P_ERROR_TEXT	IN	VARCHA R2	No	Error text to be passed when scheduling a Unit Section Occurrence fails
P_ORG_ID	IN	NUMBER	No	Operating unit identifier

Table 19–3 Scheduling API insert procedure parameters

Parameter Name	IN/ OUT	Data Type	Required	
X_RETURN_STATUS	OUT	VARCHAR2	Yes	Standard API parameter
X_MSG_COUNT	OUT	NUMBER	Yes	Standard API parameter
X_MSG_DATA	OUT	VARCHAR2	Yes	Standard API parameter

Part VII

Academic Records Module

20

Degree Audit Data Exchange Process

This chapter describes the process of transferring data from Oracle Student System to a third party degree audit application and the return process. The following sections are in this chapter:

Definition

The Degree Audit Outgoing Data process enables users to transfer information to be used for a degree audit from Oracle Student System to outgoing interface tables. This data then flows from the outgoing interface tables to a third party degree audit application.

The Degree Audit Incoming Data process enables users to transfer degree audit information from a third party degree audit application to incoming interface tables.

Overview

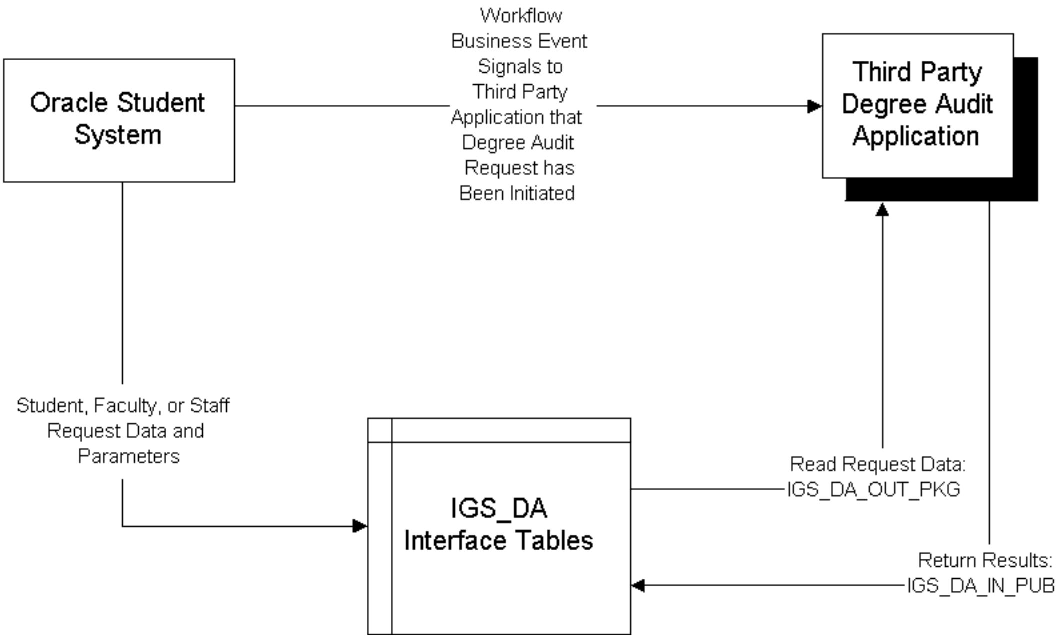
A degree audit evaluates a student’s current or expected academic progress against the institution’s requirements for an academic program.

The Degree Audit Outgoing Data process provides a method of transferring information currently stored in Oracle Student System, such as person ID, address, program, unit, and test scores, to a third party degree audit application. The Degree Audit Incoming Data process returns information from the third party degree audit application to store the data.

Degree Audit Interface Overview Flowchart

Figure 20–1, page 20-3 shows the transfer of data between Oracle Student System’s Degree Audit interface tables and a third party degree audit application as described in the accompanying text.

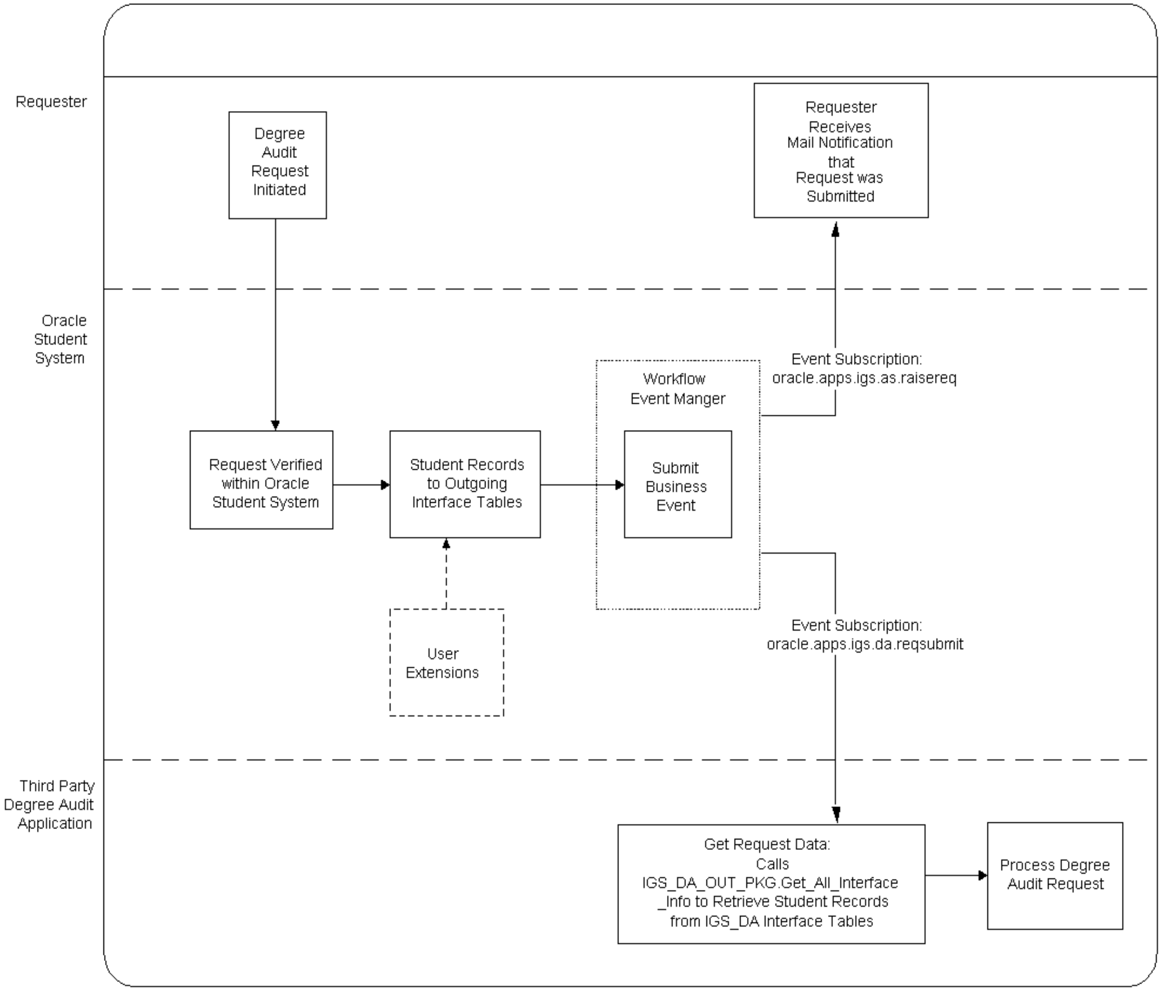
Figure 20–1 Degree Audit Interface Overview Flowchart


Oracle Student System
Workflow Business Event Signals to Third Party Application that Degree Audit Request has Been Initiated
Third Party Degree Audit Application
Student, Faculty, or Staff Request Data and Parameters
IGS_DA Interface Tables
Read Request Data: IGS_DA_OUT_PKG
Return Results: IGS_DA_IN_PUB

Degree Audit Outgoing Data Process Flowchart

Figure 20–2, page 20-4 shows the Degree Audit Outgoing Data process flowchart as described in the Degree Audit Outgoing Data Process section.

Figure 20–2 Degree Audit Outgoing Data Process Flowchart


Requester
Degree Audit Request Initiated
Requester Receives Mail Notification that Request was Submitted
Oracle Student System
Workflow Event Manger
Event Subscription: oracle.apps.igs.as.raisereq
Request Verified within Oracle Student System
Student Records to Outgoing Interface Tables
Submit Business Event
User Extensions
Event Subscription: oracle.apps.igs.da.reqsubmit
Third Party Degree Audit Application
Get Request Data: Calls IGS_DA_OUT_PKG.Get_All_Interface _Info to Retrieve Student Records from IGS_DA Interface Tables
Process Degree Audit Request

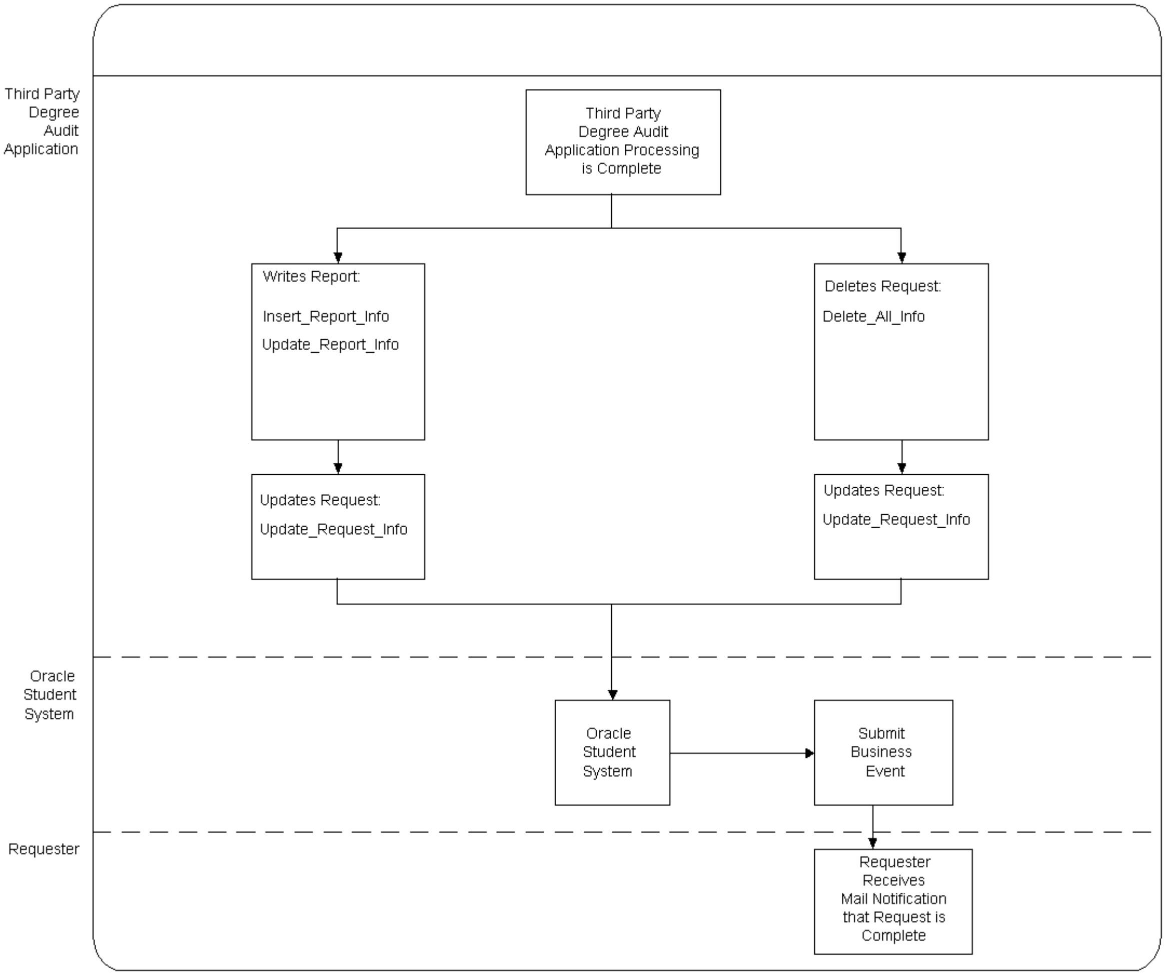
Degree Audit Outgoing Data Process

1. A student, faculty, or staff member, such as an academic advisor or a representative from the Admissions or Registrar's Office, initiates the Degree Audit process request from Oracle Student System self service.
2. Oracle Student System verifies the request and populates the Degree Audit outgoing interface tables.
3. For population of the outgoing interface tables, extensions enable users to customize data in the request, user definition, student, and program tables.
4. After outgoing interface tables are populated, Oracle Workflow submits a business event, which signals to a third party degree audit application that a Degree Audit request has been initiated.
5. Subscribers to the business event receive notification that the event has occurred.
   - The Degree Audit requester receives notification that the request is submitted.
   - The third party degree audit application's event subscription captures the batch ID of the Degree Audit Outgoing process request from the Workflow business event and initiates the data retrieval process by calling the public package IGS_DA_OUT_PKG.Get_All_Interface_Info to retrieve the data from the IGS_DA interface tables for the particular batch ID.

Degree Audit Incoming Data Process Flowchart

Figure 20–3, page 20-6 shows the Degree Audit Incoming Data process flowchart as described in the Degree Audit Incoming Data Process section.

Figure 20–3 Degree Audit Incoming Data Process Flowchart



Degree Audit Incoming Data Process

Once the processing of the Degree Audit data request is complete, the third party degree audit application has two options for returning information to Oracle Student System:

- returning Degree Audit results
- deleting a request

Returning Degree Audit Results

1. Once the Degree Audit outgoing data request is complete, the third party degree audit application calls the public package IGS_DA_IN_PUB.Insert_Report_Info to write the report as a CLOB, Character Large Object, into Oracle Student System.
2. The third party degree audit application then calls the public package IGS_DA_IN_PUB.Update_Request_Info that informs Oracle Student System that the degree audit request has been completed by the third party degree audit application by updating the batch request status.

   This process occurs whether or not a report is returned.

Deleting a Request

1. If the request processing has raised an exception or if the request is declined, the third party degree audit application calls the public package IGS_DA_IN_PUB.Delete_All_Info to delete all records in the IN tables for a particular batch ID.

   The IN table information is data produced by the third party application.
2. The third party degree audit application then calls the public package IGS_DA_IN_PUB.Update_Request_Info that informs Oracle Student System that the degree audit request has been completed by the third party degree audit application by updating the batch request status.

   This process occurs whether or not the IN table information is deleted.

Interface Tables

Oracle Student System uses the following outgoing Degree Audit interface tables during the Degree Audit Outgoing Data process:

- IGS_DA_CNFG_SRT_ORD
- IGS_DA_OUT_PRG
- IGS_DA_OUT_PRG_CGRY
- IGS_DA_OUT_PRG_REF
- IGS_DA_OUT_PRG_USET
- IGS_DA_OUT_SPCL_REQ
- IGS_DA_OUT_STDNT
- IGS_DA_OUT_STDNT_DEG
- IGS_DA_OUT_STDNT_SPT
- IGS_DA_OUT_TST_SGMT
- IGS_DA_OUT_TSTS
- IGS_DA_OUT_UNT_AH
- IGS_DA_OUT_UNT_AS
- IGS_DA_OUT_UNT_CAT
- IGS_DA_OUT_UNT_EN
- IGS_DA_OUT_UNT_REF
- IGS_DA_OUT_USR_DEF
- IGS_DA_RQST

For information on table column descriptions and values, see Oracle Student System Table Information at http://metalink.oracle.com/metalink/plsql/ml2_gui.startup.

Part VIII

HESA Data Import

21

HESA Data Import Process

This chapter describes the HESA data import process. The following sections are in this chapter:

Definition

Institutions in the UK must report student data to Higher Education Statistics Agency (HESA). Institutions can use HESA information of new students provided by Universities and Colleges Administration Service (UCAS) in the *J file for reporting to HESA. A description of the file format is available from UCAS.

The information provided in the *J file is imported into Oracle Student System through the HESA data import process.

Overview

HESA data is loaded into an interface table, from which it is later uploaded into Oracle Student System through a concurrent process.

The source data to populate the interface table can come from various external sources including UCAS and legacy systems.

The following information is provided by UCAS to the institutions in the fixed-format file.

- personal information including surname, given names, gender, date of birth, special need, ethnicity, and nationality
- the person's UCAS, HESA and Scottish Candidate number alternate person identification numbers
- information about the entrant’s previous education including previous institution, date of leaving the previous institution, highest qualification on entry, and UCAS tariff
- monitoring information including domicile code, social class, and parents’ occupations
- the subjects of qualification aim and proportion derived from the UCAS course code on to which the entrant was accepted
- fee eligibility

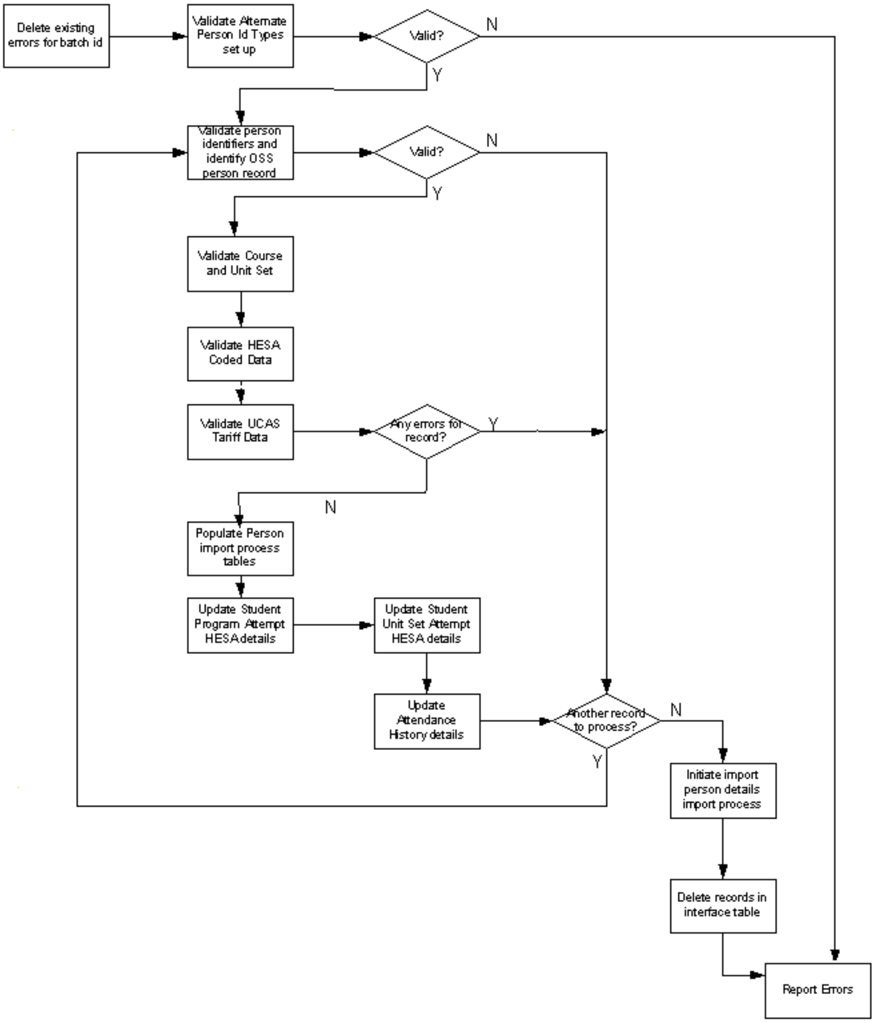
References

For information on HESA reporting, see *Oracle Student System for the United Kingdom User’s Guide*.

HESA Data Import Process Functional Flowchart

Figure 21–1, page 21-3 shows the HESA Data Import process.

Figure 21–1 HESA Data Import Process Functional Flowchart


Delete existing errors for batch id
Validate Alternate Person Id Types set up
Valid?
N
Y
Validate person identifiers and identify OSS person record
Valid?
N
Y
Validate Course and Unit Set
Validate HESA Coded Data
Validate UCAS Tariff Data
Any errors for record?
Y
N
Populate Person import process tables
Update Student Program Attempt HESA details
Update Student Unit Set Attempt HESA details
Update Attendance History details
Another record to process?
N
Y
Initiate import person details import process
Delete records in interface table
Report Errors

HESA Data Import Process

Importing HESA data involves loading students' details into interface table, and then running a concurrent process to upload the data into Oracle Student System.

Prerequisites

The following code types and associations must be set up in Maintain Codes and Maintain Code Mappings windows before starting the HESA data import process:

- Oracle Student System Person ID HUSID must exist.
- Oracle Student System Person ID Type UCASID must exist.
- Oracle Student System Person ID Type UCASREGNO must exist.
- Oracle Student System Alternate organization ID Type HESA_INST must exist.
- UCAS tariff calculation type must be specified as external calculation if you wish to import UCAS tariff data. You can specify this in the UCAS Tariff Calculation Setup window.

  For information on setting up code types and associations, and on UCAS tariff calculation setup, see *Oracle Student System Implementation Guide*.

Note: The entrants included in the file must already exist in Oracle Student System and they must have existing student program attempt, student unit set attempt, and the corresponding HESA records. This process does not attempt to create and insert new records where these do not exist.

Steps

The HESA data import process is outlined below.

1. Institutions create a batch definition record in the IGS_HE_BATCH_INT table before entering data from that batch into the interface table.
2. Institutions enter data into the interface table IGS_HE_UCAS_IMP_INT provided by Oracle Student System. The institutions must devise their own means of entering data into the interface tables.

   Institutions can choose which of the data items provided by UCAS to import by populating the relevant fields in the interface tables. A null field in the interface table does not overwrite data that already exists.

   Table 21–1 lists the data elements and their attributes, with comments on the data validations performed.

3. Institutions run the Import HESA Student Details concurrent process by specifying the batch ID and selecting Yes for at least one of the following parameters:

   - Import person details
   - Import HESA student program details

   For information on this concurrent process and parameters, see *Oracle Student System for the United Kingdom User's Guide.*

4. Records which are successfully uploaded are deleted from the interface table. Those which are not uploaded due to errors are left in the interface table, and they can be uploaded in another run of the concurrent process after the records are appropriately modified.

5. All records which are not uploaded into Oracle Student System tables are listed in an exception report at the end of the process.

Data Elements in the Interface Table

Table 21–1 lists the data elements to be entered into the interface table and their attributes. The comments column also provides information on the validations performed on the data element.

Table 21–1 Data elements of the HESA details interface table

Data Element Name	Data type	Length	Mandatory /Optional	Comment
Batch_id	Number	15	Mandatory	Unique ID number to identify data set; sequence number
Interface_HESA_id	Number	15	Mandatory	Sequence number: uniquely identifies each interface record across all batches.
HUSID	VARCHAR2	20	Optional	HESA unique student identifier; One of HUSID, Person Number or UCASNUM must be provided.
Person_number	VARCHAR2	30	Optional	Oracle Student System person number; One of HUSID, Person Number or UCASNUM must be provided.
Course_cd	VARCHAR2	6	Optional	Oracle Student System Program code; If this is provided a valid HESA student program attempt details record must exist.

Table 21–1 Data elements of the HESA details interface table

Data Element Name	Data type	Length	Mandatory /Optional	Comment
Unit_set_cd	VARCHAR2	10	Optional	Oracle Student System unit set code; If this is provided a valid HESA student unit set attempt details record must exist.
UCASNUM	VARCHAR2	20	Optional	UCAS applicant number; One of HUSID, Person Number or UCASNUM must be provided.
SCOTVEC	VARCHAR2	20	Optional	Scottish candidate number.
Surname	VARCHAR2	150	Optional	Student's surname
Given_names	VARCHAR2	150	Optional	Student's given names
Birth_Dt	Date		Optional	Student's date of birth
Sex	VARCHAR2	30	Optional	HESA gender code: this must map to a valid Oracle Student System gender code under the association code UC_OSS_HE_ GEN_ASSOC.
Domicile_cd	VARCHAR2	30	Optional	HESA country code: this must map to a valid user defined Oracle Student System domicile code under the association code UC_OSS_HE_ DOM_ASSOC.
County_code	VARCHAR2	30	Optional	HESA country code: this must map to a valid Oracle Student System TCA Country code under the association code UC_OSS_HE_ NAT_ASSOC.
Ethnic_origin	VARCHAR2	60	Optional	HESA ethnicity code: this must map to a valid user-defined Oracle Student System ethnicity code under the association code UC_OSS_HE_ ETH_ASSOC.
Disability_type	VARCHAR2	10	Optional	HESA disability code: this must map to a valid user defined Oracle Student System special needs type code under the association code UC_OSS_HE_DIS_ASSOC.
Year_left	NUMBER	4	Optional	Year the student left the previous institution spelled out as YYYY. Year_left can be specified only when Inst_code is also specified.
Occcode	VARCHAR2	30	Optional	The pre-2000 parental occupation code

Table 21–1 Data elements of the HESA details interface table

Data Element Name	Data type	Length	Mandatory /Optional	Comment
Highest_qual_on_ entry	VARCHAR2	30	Optional	HESA qualification-on-entry code: this must be a valid grade within the HESA highest qualification on entry grading schema which is set up as a value under code type HESA_HIGH_QUAL_ON_ENT.
Subject_qualaim1	VARCHAR2	10	Optional	JACS code: this must be a valid Oracle Student System government field of study code.
Subject_qualaim2	VARCHAR2	10	Optional	JACS code: this must be a valid Oracle Student System government field of study code.
Subject_qualaim3	VARCHAR2	10	Optional	JACS code: this must be a valid Oracle Student System government field of study code.
Qualiaim proportion	VARCHAR2	30	Optional	HESA SBJBID code: this must map to a valid user defined subject proportion code under the association code OSS_HESA_PROPORTION.
Fee_eligibility	VARCHAR2	30	Optional	HESA fee eligibility code: this must map to a valid user defined fee eligibility code under the association code OSS_HESA_ FEEELIG_ASSOC.
Postcode	VARCHAR2	30	Optional	Student's post code
Social_class_ind	VARCHAR2	30	Optional	HESA code: this must map to a valid user defined social class indicator under the association code UC_OSS_HE_ SOC_ASSOC.
Occupation_code	VARCHAR2	30	Optional	HESA code: this must map to a valid user-defined parental occupation code under the association code US_OSS_HE_ OCC_ASSOC.
Inst_code	VARCHAR2	30	Optional	Institution code: this must be a valid alternate institution code with alternate institution type of HESA_INST. This must be specified if year_left is specified.
GCEASN	Number	2	Optional	Number of GCE AS levels: this is mandatory if GCEAST is provided. A valid Oracle Student System award code must be mapped to UCAS tariff level GCSEAS under the external calculation type.
GCEASTS	Number	3	Optional	GCE AS Levels tariff score: this is mandatory if GCEASN is provided.

Table 21–1 Data elements of the HESA details interface table

Data Element Name	Data type	Length	Mandatory /Optional	Comment
VCEASN	Number	2	Optional	Number of VCE AS levels: this is mandatory if VCEAST is provided. A valid Oracle Student System award code must be mapped to UCAS tariff level VCSEAS under the external calculation type.
VCEASTS	Number	3	Optional	VCE AS levels tariff score: this is mandatory if VCEASN is provided
GCEAN	Number	2	Optional	Number of GCE A levels: this is mandatory if GCEAT is provided. A valid Oracle Student System award code must be mapped to UCAS tariff level GCSEA under the external calculation type.
GCEATS	Number	3	Optional	GCE A levels tariff score: this is mandatory if GCEAN is provided.
VCEAN	Number	2	Optional	Number of VCE A levels: this is mandatory if VCEAT is provided. A valid Oracle Student System award code must be mapped to UCAS tariff level VCSEA under the external calculation type.
VCEATS	Number	3	Optional	VCE A Levels tariff score: this is mandatory if VCEAN is provided.
KSQN	Number	2	Optional	Number of key skills: this is mandatory if KSQTS is provided. A valid Oracle Student System award code must be mapped to UCAS tariff level KEYSKL under the external calculation type.
KSQTS	Number	3	Optional	Key skills tariff score: this is mandatory if KSQN is provided.
UKSAN	Number	2	Optional	Number of 1 unit key skills: this is mandatory if UKSATS is provided. A valid Oracle Student System award code must be mapped to UCAS tariff level 1UNKEYSKL under the external calculation type.
UKSATS	Number	3	Optional	One unit key skills tariff score: this is mandatory if UKSAN is provided.

Table 21–1 Data elements of the HESA details interface table

Data Element Name	Data type	Length	Mandatory /Optional	Comment
SAHN	Number	2	Optional	Number of advanced highers: this is mandatory if SAHTS is provided. A valid Oracle Student System award code must be mapped to UCAS tariff level SCOTDH under the external calculation type.
SAHTS	Number	3	Optional	Advanced highers tariff score: this is mandatory if SAHN is provided.
SHN	Number	2	Optional	Number of highers: this is mandatory if SHTS is provided. A valid Oracle Student System award code must be mapped to UCAS tariff level SCOTH under the external calculation type.
SHTS	Number	3	Optional	Number of highers tariff score: this is mandatory if SHN is provided.
SI2N	Number	2	Optional	Number of intermediates: this is mandatory if S12TS is provided. A valid Oracle Student System award code must be mapped to UCAS tariff level SCOTI2 under the external calculation type.
SI2TS	Number	3	Optional	Intermediates tariff score: this is mandatory if S12N is provided.
SSGCN	Number	2	Optional	Number of standard grade: this is mandatory if SSGCTS is provided. A valid Oracle Student System award code must be mapped to UCAS tariff level SCOTST under the external calculation type.
SSGCTS	Number	3	Optional	Standard grade tariff score: this is mandatory if SSGCN is provided.
SCSN	Number	2	Optional	Number of Scottish core skills: this is mandatory if SCSTS is provided. A valid Oracle Student System award code must be mapped to UCAS tariff level SCOTCO under the external calculation type.
SCSTS	Number	3	Optional	Scottish core skills tariff score: this is mandatory if SCSN is provided.

Table 21–1 Data elements of the HESA details interface table

Data Element Name	Data type	Length	Mandatory /Optional	Comment
AEAN	Number	2	Optional	Number of advanced extension awards: this must be specified if AETS is specified. A valid Oracle Student System award code must be mapped to UCAS tariff level ADVEXT under the external calculation type.
AEATS	Number	3	Optional	Advanced extension awards tariff score: this must be specified if AEAN is specified.
Total_UCAS_tariff	Number	3	Optional	Total UCAS tariff score. Total tariff cannot be specified when all of the UCAS tariff score fields are null.

Interface Tables

Oracle Student System uses the following interface tables during the HESA data import process:

- IGS_HE_BATCH_INT
- IGS_HE_UCAS_IMP_INT
- IGS_HE_UCAS_IMP_ERR

For information on table column descriptions and values, see Oracle Student System Table Information at http://metalink.oracle.com/metalink/plsql/ml2_gui.startup.

Part IX

References

A

Admission Import Process Error Codes

This appendix describes the Oracle Student System import process error codes.

Import Process Error Codes

Institutions must verify the validity of data elements before importing data into Oracle Student System to avoid errors. Error codes are used to update the interface table affected by the error condition. An error report is generated indicating why the import process was interrupted. The error may be caused by incorrect data or may be due to system fatal errors. The error record is skipped and the next record is processed.

Table A, page A-1, describes each of the error codes that may be generated.

Table A Import Process Error Codes

Import Error Code	Meaning
E001	no records found to import
E002	more than one duplicate person exists
E003	partial matched persons exist
E004	preferred alternate person ID insertion failed
E005	person statistics insertion failed
E006	person address insertion failed
E007	alternate person ID insertion failed
E008	validation of the field failed
E009	test results insertion failed

Table A Import Process Error Codes

Import Error Code	Meaning
E010	person type insertion failed
E011	athletic details insertion failed
E012	application insertion failed
E013	relationship exists
E014	update failed
E015	academic details insertion failed
E016	record discrepancy exists
E017	duplicate criteria fields do not exist
E018	person number validation failed
E019	person identifier validation failed
E020	educational goal details record already exists
E021	admission application number cannot be null
E022	could not update application offer
E023	sequence number cannot be null
E024	record locked, cannot update application instance
E025	application document, entry qualification status not updated
E026	item level validation failed
E027	admission outcome status validation failed
E028	admission offer deferment status validation failed
E029	conditional offer validation failed
E030	admission course not valid for an offer
E031	invalid admission offer date
E032	invalid offer response date
E033	invalid admission course offering set
E034	could not update application outcome status as OFFER
E035	unable to retrieve the mandatory steps for admission offer
E036	mandatory academic history not populated

Table A Import Process Error Codes

Import Error Code	Meaning
E037	mandatory test details not populated
E038	mandatory extra-curricular activity details not populated
E039	mandatory recruitment details not populated
E040	mandatory educational goals details not populated
E041	mandatory special interest details not populated
E042	mandatory special talents details not populated
E043	mandatory credentials details not populated
E044	mandatory athletics details not populated
E045	mandatory other institution details not populated
E046	mandatory academic interest details not populated
E047	mandatory application intent details not populated
E048	mandatory rating indicator details not populated
E049	mandatory program approval indicator details not populated
E050	mandatory application note details not populated
E051	multiple offer validation failed
E052	honor date should be within birth date and system date
E053	recruitment validation failed - error inserting a record
E054	recruitment validation failed - person ID
E055	recruitment validation failed - special interest ID
E056	recruitment validation failed - Very Important Person
E057	recruitment validation failed - deactivate recruitment status
E058	recruitment validation failed - program interest ID
E059	recruitment validation failed - institution size ID
E060	recruitment validation failed - institution control ID
E061	recruitment validation failed - institution setting ID
E062	recruitment validation failed - institution location ID
E063	recruitment validation failed - special services ID

Table A Import Process Error Codes

Import Error Code	Meaning
E064	recruitment validation failed - employment ID
E065	recruitment validation failed - housing ID
E066	recruitment validation failed - degree goal ID
E067	recruitment validation failed - unit set ID
E068	recruitment validation failed - certainty of choice ID
E069	recruitment validation failed - religion ID
E070	recruitment validation failed - secondary school location ID
E071	could not update person recruitment record
E072	academic calendar in interface table and parameter are null
E075	application validation failed - athletic eligibility status
E076	application validation failed - apply for financial aid
E077	application validation failed - financial aid apply date
E078	application validation failed - application source ID
E079	application validation failed - educational goal prior to enrollment
E080	application validation failed - descriptive flexfield
E081	admission calendar in interface table and parameter are null
E082	admission calendar in interface table and parameter are null
E083	application validation failed - entry status
E084	application validation failed - entry level
E085	application validation failed - special group 1
E086	application validation failed - special group 2
E087	application validation failed - school applied to indicator
E088	application validation failed - common application
E089	rule = review: biographic and demographic updation failed
E090	rule = import: biographic and demographic updation failed
E091	exception while updating match indicator to 23
E093	athletics details insertion failed

Table A Import Process Error Codes

Import Error Code	Meaning
E094	athletics details updation failed
E095	athletics details validation failed: eligibility status
E096	athletics details validation failed: tentative admission code
E097	athletics details validation failed: predicted eligibility
E098	person special need (disability) validation failed: disability type
E099	athletics program insertion failed
E100	athletics program updation failed
E101	athletics program validation failed: athletic program code
E102	athletics program validation failed: end date
E103	athletics program validation failed: recruited indicator
E104	athletics program validation failed: participating indicator
E105	biographic and demographic details validation failed: country
E106	biographic and demographic details validation failed: state
E107	biographic and demographic details validation failed: citizenship status
E108	biographic and demographic details validation failed: start date
E109	housing status insertion failed
E110	housing status insertion failed: teaching period house status
E111	housing status insertion failed: end date
E112	housing status validation failed: mutually inclusive calendar type, sequence number
E113	housing status validation failed: calendar type and sequence number
E114	housing status updation failed
E115	felony details validation failed: convict indicator
E116	hearing details validation failed: disciplinary file indicator
E117	hearing details validation failed: academic dismissal indicator
E118	hearing details validation failed: non-academic dismissal indicator
E119	hearing details validation failed: start date greater than end date

Table A Import Process Error Codes

Import Error Code	Meaning
E120	felony details insertion failed
E121	felony details insertion failed
E122	hearing details insertion failed
E123	hearing details updation failed
E124	international details validation failed: fund source
E125	international details validation failed: country
E126	international details validation failed: document type
E127	international details: insert into HZ table failed
E128	international details: update into HZ table failed
E129	rule I; person citizenship updation failed
E130	rule R; person citizenship updation failed
E132	fund source validation failed: document indicator
E133	fund source insertion failed
E134	fund dependants insertion failed
E135	fund source validation failed: party relationship type code
E136	rule I; fund dependants updation failed
E137	rule R; fund dependants updation failed
E138	fund dependants: exception while updating Match Ind = 23
E139	person special needs (disability) validation failed: early registration indicator
E140	person special needs (disability) validation failed: special allowance
E141	person special needs (disability) validation failed: support level
E142	person special needs (disability) validation failed: end date
E143	person special needs (disability) validation failed: descriptive flexfield
E144	person special needs (disability) validation failed: interviewer ID
E145	person special needs (disability) insertion failed
E146	person special needs (disability) updation failed
E147	person special needs (disability) exception from child (SERVICE)

Table A Import Process Error Codes

Import Error Code	Meaning
E148	person special needs (disability) exception from child (CONTACT)
E149	person special needs (service) validation failed: special service code
E150	person special needs (service) validation failed: documented indicator
E151	person special needs (service) insertion failed
E152	person special needs (service) updation failed
E153	person special needs (contact) insertion failed
E154	person special needs (contact) updation failed
E155	person special needs no child process, DISABILITY_ID = NULL
E156	immunization details validation failed: immunization code
E157	immunization details validation failed: status code
E158	immunization details validation failed: start date greater than end date
E159	immunization details insertion failed
E160	immunization details updation failed
E161	statistics details (biographic and demographic) insertion failed
E162	valid evaluator exists
E163	valid prospect exists
E170	person statistics descriptive flexfield validation failed
E171	relationship code is NULL
E172	relationship creation failed
E173	phone country code validation failed
E174	veteran status validation failed
E201	validation failed - Pre Name adjunct
E202	validation failed - sex
E203	validation failed - birth date
E204	validation failed - person number cannot be null
E205	validation failed - religion ID
E206	validation failed - marital status

Table A Import Process Error Codes

Import Error Code	Meaning
E207	validation failed - ethnic origin
E208	start date should be less than end date
E209	validation failed - country
E210	person information not present
E211	validation failed - site use code
E212	validation failed - start date is required
E213	value should be in Y or N
E214	validation failed - college activity type
E215	validation failed - activity source code is required
E216	validation failed - interest type code is required
E217	validation failed - program code
E218	start date should be less than current/system date
E219	validation failed - weeks per year
E220	value should be less than zero
E221	validation failed - alias type inactive
E222	start date should not be less than birth date of person
E223	validation failed - occupational title code
E224	validation failed - type of employment
E225	validation failed - fraction of employment
E226	validation failed - tenure of employment
E227	validation failed - weekly work hours
E228	validation failed - overlapping dates
E229	validation failed - level of interest
E230	validation failed - activity source code
E231	validation failed - sub interest type code
E232	validation failed - interest type code
E233	validation failed - level of participation

Table A Import Process Error Codes

Import Error Code	Meaning
E234	validation failed - sport indicator
E235	validation failed - relationship code
E236	validation failed - visa type
E237	Visa Issue Date should be less than Entry Date
E238	visa expiry date should be greater than entry date
E239	Validation Failed - Relationship Type
E240	validation failed - other relationship type
E241	deceased date cannot be greater than the current date
E242	visa expiry date should be greater than issue date
E243	given relationship information already exists
E244	insertion failed - address usage
E245	updation failed - address usage
E246	validation failed - contact type
E247	validation failed - phone line type
E248	validation failed- e-mail format
E249	validation failed - phone country code
E250	phone number or phone line type cannot be null
E251	contact type is e-mail, e-mail address cannot be null
E252	program code and version number combination invalid
E253	validation failed - current institution
E254	validation failed - sub interest type code and interest type code mismatch
E255	validation failed - descriptive flexfield
E256	validation failed - college activity is required
E257	validation failed - date recognized is required
E258	alternate person ID - person ID type validation failed
E259	validation failed - visa number is required
E260	visa expiry date cannot be provided, visa issue date is null

Table A Import Process Error Codes

Import Error Code	Meaning
E261	entry date cannot be provided, visa issue date is null
E262	visa issue date cannot be less than birth date of person
E263	entry date cannot be provided, visa expiry date is null
E264	validation failed - international agent organizational unit code
E265	validation failed - international agent person id
E266	passport expiry date cannot be less than birth date of person
E267	disowned date should lie between recognized and end date
E268	Alternate Person ID validation failed
E269	none disability type cannot be created
E270	disability type cannot be created
E271	special needs service cannot be created
E272	special needs contact cannot be created
E300	person ID type not created
E301	validation failed - teaching period
E302	validation failed - inquiry status
E303	validation failed - inquiry source type
E304	validation failed - entry status ID
E305	validation failed - entry level ID
E306	validation failed - registering person ID
E307	validation failed - override process indicator
E308	validation failed - educational goal ID
E309	validation failed - school of interest ID
E310	validation failed - learn source ID
E311	validation failed - influence ID
E312	inquiry program code ID does not exist for the entry status ID
E315	validation failed - program code ID
E316	unit set code ID does not exists for the entry status ID

Table A Import Process Error Codes

Import Error Code	Meaning
E317	validation failed - unit set code ID
E318	validation failed - information type
E319	validation failed - characteristic type
E320	inquiry package item does not exist for the entry status
E321	validation failed - package item
E322	insert failed
E323	validation failed - unit difficulty
E324	validation failed - credit points attempted
E325	validation failed - credit points earned
E326	validation failed - start date
E327	validation failed - end date
E328	validation failed - total credit points attempted
E329	validation failed - total credit points earned
E330	validation failed - total unit grade points
E331	validation failed - total grading point average units
E332	validation failed - transcript status
E333	validation failed - transcript type
E334	validation failed - transcript source
E335	validation failed - date of receipt
E336	validation failed - converted GPA
E337	validation failed - converted grading schema ID
E338	validation failed - term type
E339	validation failed - approximate rank
E340	validation failed - weighted rank
E341	validation failed - rank in class
E342	validation failed - class size
E343	validation failed - quartile rank

Table A Import Process Error Codes

Import Error Code	Meaning
E344	validation failed - quintile rank
E345	validation failed - decile rank
E346	validation failed - percentile rank
E347	error occurred while processing a child record
E348	validation failed - date of issue
E401	validation failed - INSTITUTION_CODE
E402	validation failed - PROGRAM_TYPE_ATTEMPTED
E403	validation failed - PROGRAM_TYPE_EARNED
E404	validation failed - PROGRAM_CODE
E405	validation failed - START_DATE
E406	validation failed - END_DATE
E407	Validation Failed - Start Date cannot be NULL
E408	Validation Failed - PLANNED_COMPLETION_DATE
E409	Validation Failed - SELFREP_TOTAL_CP_ATTEMPTED
E410	Validation Failed - SELFREP_TOTAL_CP_EARNED
E411	validation failed - SELFREP_TOTAL_GP_UNITS_ATTEMPTE
E412	validation failed - SELFREP_INST_GPA
E413	validation failed - SELFREP_GRADING_SCALED_ID
E414	validation failed - SELFREP_WEIGHTED_GPA
E415	validation failed - SELFREP_RANK_IN_CLASS
E416	validation failed - SELFREP_WEIGHTED_RANK
E417	validation failed - CLASS_SIZE
E418	validation failed - flexfield - academic history
E419	validation failed - TRANSCRIPT_REQUIRED
E420	validation failed - FIELD_OF_STUDY
E421	Validation Failed - ACAD_HONOR_TYPE_ID
E422	validation failed - INTENT_TYPE_ID

Table A Import Process Error Codes

Import Error Code	Meaning
E423	validation failed - Flexfield - Applicant Intent
E424	validation failed - NOTE_TYPE_ID
E425	validation failed - INSTITUTION_CD
E426	validation failed - SPECIAL_INTEREST_TYPE_ID
E427	validation failed - SPECIAL_TALENT_TYPE_ID
E430	validation failed - ADMISSION_TEST_TYPE
E431	validation failed - test_date
E432	validation failed - SCORE_REPORT_DATE
E433	validation failed - EDU_LEVEL_ID
E434	validation failed - SCORE_TYPE
E435	validation failed - SCORE_SOURCE_ID
E436	validation failed - NON_STANDARD_ADMIN
E437	validation failed - COMP_TEST_SCORE
E438	validation failed - PERCENTILE
E439	validation failed - NATIONAL_PERCENTILE
E440	validation failed - STATE_PERCENTILE
E441	validation failed - TEST_SEGMENT_ID
E442	validation failed - IRREGULARITY_CODE
E443	validation failed - TEST_SCORE
E444	validation failed - PERCENTILE_YEAR_RANK
E445	validation failed - SCORE_BAND_LOWER
E446	validation failed - SCORE_BAND_UPPER
E447	validation failed - SPECIAL_CODE
E450	validation failed - primary flag value should be Y or N
E451	validation failed - deceased date
E452	relation person creation failed
E453	validation failed - current institution should be Y or N

Table A Import Process Error Codes

Import Error Code	Meaning
E501	decision maker ID validation failed
E502	decision date validation failed
E503	decision reason validation failed
E504	pending reason validation failed
E505	offer date validation failed
E506	offer response date validation failed
E507	documentation status not completed
E508	entry qualification status not qualified
E509	withdrawn application cannot be updated
E510	program offering option validation failed
E511	admission outcome status validation failed
E512	validation of outcome status update failed
E513	admission conditional offer validation failed
E514	offer date validation failed
E515	offer response date validation failed
E516	admission decision details validation failed
E517	admission outcome status update failed
E518	unhandled exception
E519	setting of end date to null for an existing prospect type failed
E520	setting of end date for an existing prospect type failed
E521	replacement of Other Person Type with a new Person Type failed
E522	person type and funnel status could not be retrieved
E523	update admission application number not found in IGS_AD_APPL
E524	validation failed for unit set code column
E525	combination of unitset and rank column already present

Table A Import Process Error Codes

Import Error Code	Meaning
E526	validation failed for education goal ID column
E527	no application instance imported successfully
E528	person is deceased

B

Admission Import Source Category Tables

This appendix describes the Oracle Student System source category tables. The following sections are in this appendix:

- Source Categories
- Source Category Attributes

Source Categories

Table Table B-1, page B-2 describes the source categories that are imported for the system source type selected for import. For example, if the system source type selected is PROSPECT LIST, then the PERSON_DETAILS source category must be imported, but all other source categories are optional. Source categories with the mandatory indicator set to N are optional when importing source types.

Table Table B-1, page B-2 also lists the complete set of categories that Oracle Student System may capture for different source types as APPLICATION or PROSPECT LIST. Mandatory elements are indicated in the Mandatory column as Y. Values for optional categories can be manually entered, taken from internal sources, or imported through an interface table.

Table B-1 Source Categories

SYSTEM SOURCE TYPE	SOURCE CATEGORY	MANDATORY	TABLE NAME
APPLICATION	PERSON	Y	IGS_AD_INTERFACE_ALL
	PERSON_STATISTICS	Y	IGS_AD_STAT_INT_ALL
	PERSON_ACADEMIC_HISTORY	N	IGS_AD_ACADHIS_INT_ALL
	PERSON_ADDRESS	N	IGS_AD_ADDR_INT_ALL
		N N	IGS_AD_ADDRUSAGE_INT_ALL
	PERSON_ALIAS	N	IGS_AD_ALIAS_INT_ALL
	PERSON_ATHLETICS	N	IGS_PE_ATH_DTL_INT
		N	IGS_PE_ATH_PRG_INT
	PERSON_CONTACTS	N	IGS_AD_CONTACTS_INT_ALL
	PERSON_DISCIPLINARY_DTLS	N	IGS_PE_FLNY_DTL_INT
		N	IGS_PE_HEAR_DTL_INT
	PERSON_EMPLOYMENT_DETAILS	N	IGS_AD_EMP_INT_ALL

Table B-1 Source Categories

SYSTEM SOURCE TYPE	SOURCE CATEGORY	MANDATORY	TABLE NAME
	PERSON_ACTIVITIES	N	IGS_AD_EXCURR_INT_ALL
	PERSON_HEALTH_INSURANCE	N	IGS_AD_HLTH_INS_INT_ALL
		N	IGS_PE_IMMU_DTL_INT
	PERSON_HOUSING_STATUS	N	IGS_PE_HOUSING_INT
	PERSON_ID_TYPES	N	IGS_AD_API_INT_ALL
	PERSON_INTERNATIONAL_DETAILS	N	IGS_PE_VISA_INT
		N	IGS_PE_PASSPORT_INT
		N	IGS_PE_VST_HIST_INT
		N	IGS_PE_EIT_INT
		N	IGS_PE_FUND_SRC_INT
		N	IGS_PE_CITIZEN_INT
	PERSON_LANGUAGES	N	IGS_AD_LANGUAGE_INT_ALL
	PERSON_MILITARY_DETAILS	N	IGS_AD_MILITARY_INT_ALL
	PERSON_RECRUITMENT_DETAILS	N	IGS_AD_RECRUIT_INT
	PERSON_SPECIAL_NEEDS	N	IGS_AD_DISABLTY_INT_ALL
		N	IGS_PE_SN_CONCT_INT
		N	IGS_PE_SN_SRVCE_INT
	PERSON_RELATIONS	N	IGS_AD_RELATIONS_INT_ALL
	RELATIONS_ACAD_HISTORY	N	IGS_AD_RELACAD_INT_ALL
	RELATIONS_ADDRESS	N	IGS_AD_RELADDR_INT_ALL
	RELATIONS_CONTACTS	N	IGS_AD_REL_CON_INT_ALL

Table B-1 Source Categories

SYSTEM SOURCE TYPE	SOURCE CATEGORY	MANDATORY	TABLE NAME
	RELATIONS_EMPLOYMENT_DETAILS	N	IGS_AD_RELEMP_INT_ALL
TRANSCRIPTS	APPLICANT_ACADEMIC_INTERESTS	N	IGS_AD_ACADINT_INT_ALL
	PERSON_ACAD_HONORS	N	IGS_AD_ACADHONOR_INT_ALL
	APPLICANT_EDU_GOALS	N	IGS_AD_EDUGOAL_INT_ALL
	APPLICANT_INTENT	N	IGS_AD_APPINT_INT_ALL
	APPLICANT_NOTES	N	IGS_AD_NOTES_INT_ALL
	APPLICANT_OTHERINSTS_APPLIED	N	IGS_AD_OTHINST_INT_ALL
	APPLICANT_SPECIAL_INTERESTS	N	IGS_AD_SPLINT_INT_ALL
	APPLICANT_SPECIAL_TALENTS	N	IGS_AD_SPLTAL_INT_ALL
	APPLICANT_UNITSETS_APPLIED	N	IGS_AD_UNITSETS_INT_ALL
	APPLICATION	Y	IGS_AD_APL_INT
		Y	IGS_AD_PS_APPL_INST_INT
	INQUIRY_INSTANCE	N	IGS_RC_I_APPL_INT
		N	IGS_RC_I_CHAR_INT
		N	IGS_RC_I_INFO_INT
		N	IGS_RC_I_PKG_INT
	PERSON	Y	IGS_AD_INTERFACE_ALL
	PERSON_STATISTICS	Y	IGS_AD_STAT_INT_ALL
		N	IGS_PE_EIT_INT
	PERSON_ACADEMIC_HISTORY	Y	IGS_AD_ACADHIS_INT_ALL
	PERSON_ADDRESS	N	IGS_AD_ADDR_INT_ALL

Table B-1 Source Categories

SYSTEM SOURCE TYPE	SOURCE CATEGORY	MANDATORY	TABLE NAME
		N	IGS_AD_ADDRUSAGE_INT_ALL
	PERSON_ALIAS	N	IGS_AD_ALIAS_INT_ALL
	PERSON_ATHLETICS	N	IGS_PE_ATH_DTL_INT
		N	IGS_PE_ATH_PRG_INT
	PERSON_CONTACTS	N	IGS_AD_CONTACTS_INT_ALL
	PERSON_DISCIPLINARY_DTLS	N	IGS_PE_FLNY_DTL_INT
		N	IGS_PE_HEAR_DTL_INT
	PERSON_EMPLOYMENT_DETAILS	N	IGS_AD_EMP_INT_ALL
	PERSON_ACTIVITIES	N	IGS_AD_EXCURR_INT_ALL
	PERSON_HEALTH_INSURANCE	N	IGS_AD_HLTH_INS_INT_ALL
		N	IGS_PE_IMMU_DTL_INT
	PERSON_HOUSING_STATUS	N	IGS_PE_HOUSING_INT
	PERSON_ID_TYPES	N	IGS_AD_API_INT_ALL
	PERSON_INTERNATIONAL_DETAILS	N	IGS_PE_CITIZEN_INT
		N	IGS_PE_FUND_SRC_INT
	PERSON_LANGUAGES	N	IGS_AD_LANGUAGE_INT_ALL
	PERSON_MILITARY_DETAILS	N	IGS_AD_MILITARY_INT_ALL
	PERSON_RECRUITMENT_DETAILS	N	IGS_AD_RECRUIT_INT
	PERSON_SPECIAL_NEEDS	N	IGS_AD_DISABLTY_INT_ALL
		N	IGS_PE_SN_CONCT_INT
		N	IGS_PE_SN_SRVCE_INT

Table B-1   Source Categories

SYSTEM SOURCE TYPE	SOURCE CATEGORY	MANDATORY	TABLE NAME
	PERSON_RELATIONS	N	IGS_AD_RELATIONS_INT_ALL
	RELATIONS_ACAD_HISTORY	N	IGS_AD_RELACAD_INT_ALL
	RELATIONS_ADDRESS	N	IGS_AD_RELADDR_INT_ALL
	RELATIONS_CONTACTS	N	IGS_AD_REL_CON_INT_ALL
	RELATIONS_EMPLOYMENT_DETAILS	N	IGS_AD_RELEMP_INT_ALL
	INQUIRY_INSTANCE	N	IGS_RC_I_APPL_INT
		N	IGS_RC_I_CHAR_INT
		N	IGS_RC_I_INFO_INT
		N	IGS_RC_I_PKG_INT
		N	IGS_AD_INQ_PRUNT_INT
	TRANSCRIPT_DETAILS	Y	IGS_AD_TRMDT_INT_ALL
		Y	IGS_AD_TUNDT_INT_ALL
		Y	IGS_AD_TXCPT_INT_ALL
PROSPECT LIST	PERSON_DETAILS	Y	IGS_AD_INTERFACE_ALL
	ACADEMIC_HISTORY_DETAILS	N	IGS_AD_ACADHIST_INT_ALL
	ADDRESS_DETAILS	N	IGS_AD_ADDR_INT_ALL
		N	IGS_AD_ADDRUSAGE_INT_ALL
	PERSON_CONTACTS	N	IGS_AD_CONTACTS_INT_ALL
	ALIAS_DETAILS	N	IGS_AD_ALIAS_INT_ALL
	PERSON_ATHLETICS_INFORMATION	N	IGS_PE_ATH_DET_INT
		N	IGS_PE_ATH_PRG_INT

Table B-1 Source Categories

SYSTEM SOURCE TYPE	SOURCE CATEGORY	MANDATORY	TABLE NAME
	PERSON_DISCIPLINARY_DETAILS	N	IGS_PE_FLNY_DTL_INT
		N	IGS_PE_HEAR_DTL_INT
	EMPLOYMENT_DETAILS	N	IGS_AD_EMP_INT_ALL
	PERSON_ACTIVITIES	N	IGS_AD_EXCURR_INT_ALL
	HEALTH_AND_INSURANCE_DETAILS	N	IGS_AD_HLTH_INS_INT_ALL
		N	IGS_PE_IMMU_DTL_INT
	PERSON_HOUSING_STATUS	N	IGS_PE_HOUSING_INT
	PERSON_ID_TYPES	N	IGS_AD_API_INT_ALL
	INTERNATIONAL_DETAILS	N	IGS_PE_CITIZEN_INT
		N	IGS_PE_EIT_ALL
		N	IGS_PE_FUND_SRC_INT
	PERSON_LANGUAGES_DETAILS	N	IGS_AD_LANGUAGE_INT_ALL
	MILITARY_SERVICE_DETAILS	N	IGS_AD_MILITARY_INT_ALL
	RELATION_DETAILS	N	IGS_AD_RELATIONS_INT_ALL
	SPECIAL_NEEDS	N	IGS_AD_DISABLTY_INT_ALL
		N	IGS_PE_SN_CONCT_INT
		N	IGS_PE_SN_SRVCE_INT
	PERSON_STATISTICS	Y	IGS_AD_STAT_INT_ALL
	RELATIONS_ACADEMIC_HISTORY_DETAILS	N	IGS_AD_RELACAD_INT_ALL
	RELATIONS_ADDRESS_DETAILS	N	IGS_AD_RELADDR_INT_ALL
	RELATIONS_CONTACTS	N	IGS_AD_REL_CON_INT_ALL

Table B-1 Source Categories

SYSTEM SOURCE TYPE	SOURCE CATEGORY	MANDATORY	TABLE NAME
TEST RESULTS	RELATIONS_EMPLOYMENT_DETAILS	N	IGS_AD_RELEMP_INT_ALL
	PERSON_DETAILS	Y	IGS_AD_INTERFACE_ALL
	ACADEMIC_HISTORY_DETAILS	N	IGS_AD_ACADHIST_INT_ALL
	ADDRESS_DETAILS	N	IGS_AD_ADDR_INT_ALL
		N	IGS_AD_ADDRUSAGE_INT_ALL
	ALIAS_DETAILS	N	IGS_AD_ALIAS_INT_ALL
	PERSON_ATHLETICS_INFORMATION	N	IGS_PE_ATH_DET_INT
		N	IGS_PE_ATH_PRG_INT
	PERSON_CONTACTS	N	IGS_AD_CONTACTS_INT_ALL
	PERSON_DISCIPLINARY_DETAILS	N	IGS_PE_FLNY_DTL_INT
		N	IGS_PE_HEAR_DTL_INT
	EMPLOYMENT_DETAILS	N	IGS_AD_EMP_INT_ALL
	PERSON_ACTIVITIES	N	IGS_AD_EXCURR_INT_ALL
	HEALTH_AND_INSURANCE_DETAILS	N	IGS_AD_HLTH_INS_INT_ALL
	PERSON_HOUSING_STATUS	N	IGS_PE_HOUSING_INT
	PERSON_ID_TYPES	N	IGS_AD_API_INT_ALL
	PERSON_INTERNATIONAL_DETAILS	N	IGS_PE_VISA_INT
		N	IGS_PE_PASSPORT_INT
		N	IGS_PE_VST_HIST_INT

Table B-1 Source Categories

SYSTEM SOURCE TYPE	SOURCE CATEGORY	MANDATORY	TABLE NAME
		N	IGS_PE_EIT_INT
		N	IGS_PE_FUND_SRC_INT
		N	IGS_PE_CITIZEN_INT
	PERSON_LANGUAGES_DETAILS	N	IGS_AD_LANGUAGE_INT_ALL
	MILITARY_SERVICE_DETAILS	N	IGS_AD_MILITARY_INT_ALL
	RELATION_DETAILS	N	IGS_AD_RELATIONS_INT_ALL
	SPECIAL_NEEDS	N	IGS_AD_DISABLTY_INT_ALL
		N	IGS_PE_SN_CONCT_INT
		N	IGS_PE_SN_SRVCE_INT
	PERSON_STATISTICS	Y	IGS_AD_STAT_INT_ALL
	RELATIONS_ACADEMIC_HISTORY_ DETAILS	N	IGS_AD_RELACAD_INT_ALL
	RELATIONS_ADDRESS_DETAILS	N	IGS_AD_RELADDR_INT_ALL
	RELATIONS_CONTACTS	N	IGS_AD_REL_CON_INT_ALL
PROSPECT SELF SERVICE INQUIRY	RELATIONS_EMPLOYMENT_DETAIL S	N	IGS_AD_RELEMP_INT_ALL
	TEST_RESULTS_DETAILS	Y	IGS_AD_TEST_INT_ALL
		Y	IGS_AD_TEST_SEGS_INT_ALL
	COLLEGE_ACTIVITIES	N	IGS_AD_EXCURR_INT_ALL
	INQUIRY_DETAILS	Y	IGS_RC_I_APPL_INT
	INQUIRY_UNIT_SETS_INQUIRY	Y	IGS_IN_I_LINES_INT
	INQUIRY_PACKAGE_ITEMS	Y	IGS_RC_I_PKG_INT
	INQUIRY_INFORMATION_TYPES	Y	IGS_RC_I_INFO_INT

Table B-1 Source Categories

SYSTEM SOURCE TYPE	SOURCE CATEGORY	MANDATORY	TABLE NAME
	INQUIRY_CHARACTERISTICS	Y	IGS_RC_I_CHAR_INT
	PERSON_DISCIPLINARY_DETAILS	N	IGS_PE_FLNY_DTL_INT
		N	IGS_PE_HEAR_DTL_INT
	PERSON_HOUSING_STATUS	N	IGS_PE_HOUSING_INT
	PERSON_ATHLETICS_INFORMATION	N	IGS_AD_ATH_DTL_INT
		N	IGS_PE_ATH_DET_INT
	PERSON_DETAILS	Y	IGS_AD_INTERFACE_ALL
	PERSON_STATISTICS	Y	IGS_AD_STAT_INT_ALL
	ADDRESS_DETAILS	N	IGS_AD_ADDR_INT_ALL
		N	IGS_AD_ADDRUSAGE_INT_ALL
	PERSON_CONTACTS	N	IGS_AD_CONTACTS_INT_ALL
	ACADEMIC_HISTORY_DETAILS	N	IGS_AD_ACADHIST_INT_ALL
	TEST_RESULTS_DETAILS	N	IGS_AD_TEST_INT_ALL
		N	IGS_AD_TEST_SEGS_INT_ALL
	INQUIRY_PROGRAMS	N	IGS_IN_I_LINES_INT
	EXTRACURRICULAR_ACTIVITIES	N	IGS_AD_EXCURR_INT_ALL

Source Category Attributes

Table Table B-2, page B-11 lists attributes for the person source category.

Table B-2 Person Source Category Attributes

Step Number	Source Category	Attributes
1	PERSON	BIRTH_DT
		GIVEN_NAMES
		LEVEL_OF_QUALIFICATION
		MIDDLE_NAME
		PREFERRED_GIVEN_NAME
		PREFIX
		SEX
		SUFFIX
		SURNAME
		TITLE

Index