

Oracle Fusion Cloud Human Resources

How do I configure business rules for Redwood individual compensation pages?



Oracle Fusion Cloud Human Resources

How do I configure business rules for Redwood individual compensation pages?

G42322-02

Copyright © 2021,2025, Oracle and/or its affiliates.

Author: Srinivas Vellika

Contents

Get Help	i
-----------------	----------

1 How do I configure business rules for Redwood individual compensation pages?	1
Individual Compensation Business Rules Introduction	1
Getting Started	1
Access Requirement	1
Implementation Considerations for Individual Compensation Processes	1
Configure Business Rules for Individual Compensation Processes	5
Troubleshooting Individual Compensation Business Rules	7

Get Help

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

Get Help in the Applications

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

Get Training

Increase your knowledge of Oracle Cloud by taking courses at [Oracle University](#).

Join Our Community

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

Share Your Feedback

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

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

Thanks for helping us improve our user assistance!

How do I configure business rules for Redwood individual compensation pages?

1 How do I configure business rules for Redwood individual compensation pages?

Individual Compensation Business Rules Introduction

You can extend the Redwood individual compensation pages by configuring fields and regions, changing the page properties using business rules, and creating field-level validations and tailored messages to meet your business requirements.

Business rules let you to control the display of regions and fields on Redwood pages. You can configure business rules in Oracle Visual Builder Studio Express mode to personalize Redwood pages, including defaulting and validating field values.

You can either use delivered rules built for best practices or create your own rules.

Getting Started

You need to set up Visual Builder Studio before you can start configuring business rules. For details, see [Set Up VB Studio to Extend Oracle Cloud Applications](#).

You need the Human Capital Management Application Administrator role to work in Visual Builder

Access Requirement

You need the Human Capital Management Application Administrator role to work in an Oracle Fusion apps sandbox. For details, see the [Securing HCM](#) guide.

Implementation Considerations for Individual Compensation Processes

Here are the considerations for Administer Compensation, Individual Compensation, and Personal Compensation processes.

You can typically use field value defaulting and validation in these cases.

- Default when date such that awards can be created after 1st of next quarter by employees only.
- Validate that awards can be created after 1st of next quarter by employees only.
- Validate that certain plans are excluded for line managers.

- Validate when date such that it's in the current fiscal year (1 Apr to 31 Mar).
- Validate spot bonus is effective either on 01 Apr 2024 or 01 Oct 2024.
- Validate new hire bonus doesn't exceed \$3000.
- Validate car allowance increase is below or equal to 10%.
- Validates new awards starts on or before 17th of a month.
- Validate that only one award proposal is submitted at a time.
- Validate bus and taxi reimbursement aren't claimed together.
- Validate that bonus can be allocated once in 6 months only.
- Advanced expressions working on Global HR and Recruiting pages need to be updated somewhat to work on the Administer Compensation, Individual Compensation, and Personal Compensation pages.

This table lists the supported attributes, exceptions, and the implementation recommendations for these pages:

- **My Client Groups > Administer Compensation**
- **My Client Groups > Individual Compensation**
- **My Team > Individual Compensation**
- **Me > Personal Contribution**

In the Conditions to Default Values	To Default Field Values	In the Condition to Validate Values	To Validate Field Values	Implementation Guidelines
• User Roles • Legal Employer • Business Unit • Country • Assignment attributes including ◦ Assignment Status ◦ Bargaining Unit ◦ Business Unit ◦ Department ◦ FTE ◦ Grade ◦ Grade Code ◦ Grade Ladder ◦ Hire Date ◦ Hourly or Salaried ◦ Job*	• Action Date • Exclude Plans • Exclude Options	For When Section • User Roles • Legal Employer • Business Unit • Country • Action Date For Compensation Section • User Roles • Legal Employer • Business Unit • Country • Plan • Option • Start Date • End Date	NA	• Can default Action Date only and validate individual compensation section data. • You can validate at the record or section level. ◦ Record level validations are triggered on clicking Save in the edit panel drawer. ◦ Section Level validation are triggered on clicking Continue in the process ◦ All the current allocation records will be available in the Current Allocations attribute ◦ You can iterate through the records and create rule on the above mentioned record level fields ◦ Section level rules can be written only using Advanced Expression Mode • Plan LOV and Plan – Option LOV in Business Rules is secured by the Individual Compensation Plan functional privilege: Define Variable Compensation Plan for Individual Compensation (CMP_DEFINE_VARIABLE_COMPENSATION_PLAN_FOR_INDIVIDUAL_COMPENSATION). This is required for business rule design time users to view the values in Plan LOV, Plan – Option LOV. If you don't have this security, you won't see the values in the LOV. • Initial Value is available for individual compensation fields to compare old and new values. • Current Allocations can be accessed while creating validation using Advanced Expression only. • When the flow is launched, the assignment data is fetched based on the effective date selected in Person Search. If effective date isn't selected, data will be fetched as of application date.

How do I configure business rules for Redwood individual compensation pages?

How do I configure business rules for Redwood individual compensation pages?

In the Conditions to Default Values	To Default Field Values	In the Condition to Validate Values	To Validate Field Values	Implementation Guidelines
- Job Code - Job Family* - Job Function* - LDG* - Legal Employer - Location - Next Payroll Start Date - Payroll Start Date - Person Type* - Position* - Position Budget Value* - Position Code - Position Standard Working Hours* - Primary Work Relationship - Termination Date - Working Hours - Working Hours Frequency *Can affect performance		- Value - Currency - Legal Entity - Create or Edit - Current Allocations - Prior Allocations		- Rules written for When section will always be evaluated on the effective date selected in Person Search / application date. - If Action Date is changed in When section, rules written for Compensation section will be evaluated based on the assignment data effective as of action date. - Prior Allocations: a. Similar to Current Allocations, prior allocations field is an array and user can iterate through the prior allocation records to write rule conditions. b. Because prior allocation data is constant, it will be available at both section and record level validations. c. Only when rules are written on Prior Allocations, the fragment loads prior allocation data and makes it available to the BR engine. d. Rules on Prior Allocations can be written only in Advanced Expression mode.

Field Details

Field	Field Accessor	Type	Comments
Action Date	<code>\$fields.individualCompensation.value()</code>		
Record Level Attributes			

How do I configure business rules for Redwood individual compensation pages?

How do I configure business rules for Redwood individual compensation pages?

Field	Field Accessor	Type	Comments
Plan	<code>\$fields.individualCompensation.value()</code>	Number	PlanId selected using Plan LOV which shows Plan Name
Plan Option	<code>\$fields.individualCompensation.value()</code>	Number	OptionId selected using Option LOV which shows values in the format <Plan Name> - <Option Name>
Start Date	<code>\$fields.individualCompensation.value()</code>	String	Allocation start date as string (Format: yyyy-MM-dd) Requires Advanced Expression mode for date comparison.
End Date	<code>\$fields.individualCompensation.value()</code>	String	Allocation end date as string (Format: yyyy-MM-dd) Requires Advanced Expression mode for date comparison.
Value	<code>\$fields.individualCompensation.value()</code>	String	Primary input value as string. Requires Advanced Expression mode for nonstring primary input value. For numeric field, the value should be type-casted to Number for arithmetic operations. Similarly, for date field, the value should be type-casted to Date for date comparisons.
Currency	<code>\$fields.individualCompensation.value()</code>	String	Currency for monetary primary input value
Legal Entity	<code>\$fields.individualCompensation.value()</code>	Number	LegalEntityId selected using Legal Entity LOV which shows LE Name
Create or Edit	<code>\$fields.individualCompensation.value()</code>	String	Mode attribute is to identify whether the record is newly created or an existing record.
Section Level Attributes			
Current Allocations	<code>\$fields.individualCompensation.value()</code>	Array	Object Structure <pre>[{ "PlanId": "string", "ComponentId": "string", "EffectiveStartDate": "string", "EffectiveEndDate": "string", "ValueString": "string", "DisplayCurrency": "string", "LegalEntityId": "string", "Mode": "string" }]</pre>

Field	Field Accessor	Type	Comments
			]
Prior Allocations	<code>\$fields.individualCompensationDetails.\$value()</code>	Array	Object Structure [{ "PlanId": "string", "ComponentId": "string", "EffectiveStartDate": "string", "EffectiveEndDate": "string", "ValueString": "string", "DisplayCurrency": "string", "LegalEntityId": "string", "Mode": "string" }]

Configure Business Rules for Individual Compensation Processes

Here are examples of a defaulting rule and multiple validating rules.

Default When date such that awards can be created after 1st of next quarter by employees only

Advanced Expression

```
let effectiveDate = new Date();
let currentQuarter = Math.floor((effectiveDate.getMonth() + 3) / 3);

// Get first date of current quarter
effectiveDate.setDate(1);
effectiveDate.setMonth((currentQuarter - 1) * 3);

// Use this technique for year rollover
effectiveDate.setMonth(effectiveDate.getMonth() + 3);
return effectiveDate.toISOString().slice(0, 10);
```

Validate When date such that it's in the current fiscal year (1 Apr - 31 Mar)

Advanced Expression

```
let effectiveDate = $fields.individualCompensationDetails.EffectiveDate.$value();
if (effectiveDate) {
  // Fetch current financial year start date
  let currentDate = new Date();
  let fyStartDate = null, fyEndDate = null;
  if (currentDate.getMonth() + 1 <= 3) {
    fyStartDate = new Date(currentDate.getFullYear() - 1, 3, 1);
    fyEndDate = new Date(currentDate.getFullYear(), 2, 31);
  }
}
```

```

    } else {
      fyStartDate = new Date(currentDate.getFullYear(), 3, 1);
      fyEndDate = new Date(currentDate.getFullYear() + 1, 2, 31);
    }
    return (new Date(effectiveDate) < fyStartDate || new Date(effectiveDate) > fyEndDate);
  }
  return false;

```

Validate new hire bonus doesn't exceed \$3000

Advanced Expression

```

if ($fields.individualCompensationDetails.PlanId.$value() === '100000018708168'
  && Number($fields.individualCompensationDetails.ValueString.$value()) > 3000) {
  return true;
}

return false;

```

Validate car allowance increase is below or equal to 10%

Advanced Expression

```

if ($fields.individualCompensationDetails.ComponentId.$numberValue() === 300100010424490
  && $fields.individualCompensationDetails.Mode.$initialValue() === 'EDIT') {
  let proposedValue = Number($fields.individualCompensationDetails.ValueString.$value());
  let currentValue = Number($fields.individualCompensationDetails.ValueString.$initialValue());

  if (proposedValue > (1.1 * currentValue))
    return true;
}

return false;

```

In this example, initialValue of Mode is used to handle date-effective update scenario.

For date effective update case, the new record will have Mode as 'CREATE' because it's a newly generated split.

Validate new awards start on or before 17th of a month

Advanced Expression

```

if ($fields.individualCompensationDetails.Mode.$value() == 'CREATE'
  && $fields.individualCompensationDetails.EffectiveStartDate.$value()) {

  let planSD = new Date($fields.individualCompensationDetails.EffectiveStartDate.$value());
  return (planSD.getDate() > 17)
}
return false;

```

Validate that only one award proposal is submitted at a time

Advanced Expression

```

let currentAllocations = $fields.individualCompensationDetails.CurrentAllocations.$value();
if (currentAllocations && currentAllocations.length) {
  let newAllocationCount = 0;

  for (let allocation of currentAllocations) {
    if (allocation.Mode == 'CREATE')
      newAllocationCount++;
  }
}

```

```
if(newAllocationCount > 1)
return true;
}
return false;
```

Validate bus and taxi reimbursement aren't claimed together

Advanced Expression

```
let currentAllocations = $fields.individualCompensationDetails.CurrentAllocations.$value();
if (currentAllocations && currentAllocations.length) {
  let busBonusPresent = false;
  let taxiBonusPresent = false;

  for (let allocation of currentAllocations) {
    if (allocation.PlanId == '300100113565911')
      busBonusPresent = true;

    else if (allocation.PlanId == '300100113565893')
      taxiBonusPresent = true;
  }

  return (busBonusPresent && taxiBonusPresent);
}
return false;
```

Validate bonus is allocated only once in six months

```
if ($fields.individualCompensationDetails.PlanId.$numberValue() === 300100090079227
&& $fields.individualCompensationDetails.Mode.$value() === 'CREATE') {
  let startDate = new Date($fields.individualCompensationDetails.EffectiveStartDate.$value());
  let minusSixMonth = new Date(startDate.getFullYear(), startDate.getMonth() - 6, startDate.getDay());
  minusSixMonth = minusSixMonth.toISOString().slice(0, 10);

  let priorAllocations = $fields.individualCompensationDetails.PriorAllocations.$value();
  if (priorAllocations?.length) {
    let priorSpotBonus = priorAllocations.find(record => record.PlanId == '300100004468561');
    if (priorSpotBonus?.EffectiveStartDate > minusSixMonth) {
      return true;
    }
  }
}
return false;
```

Troubleshooting Individual Compensation Business Rules

To print all the attributes for an Individual Compensation object, create a validation rule without any condition.

1. Add a message to show the proposed value by completing the fields shown here:

Field	Value
Summary	Value

Field	Value
Severity	Warning
Detail	<code>[[JSON.stringify(\$fields.individualCompensationDetails.\$value(), null, 2)]]</code>

2. Add another messages to show the initial value (the database value of the record, for EDIT case) by completing the fields shown here:

Field	Value
Summary	Initial Value
Severity	Warning To review the value of the CurrentAllocations array, set the message severity to error. Other severities let people submit or go to the next section.
Detail	<code>[[JSON.stringify(\$fields.individualCompensationDetails.\$initialValue(), null, 2)]]</code>

Note: For a new allocation, initialValue will be null because it doesn't exist in the database. The same applies for section-level validation, which runs on Continue or Submit actions.

Here's an example personal contribution showing the results both messages:

Manage Contributions



Personal Contribution

Roger ZCMP_CWB_Federer, ZCMP_CWB_CEO

+ Add

▲ Value

{ "ComponentId": "300100182420225", "DisplayCurrency": "USD", "EffectiveEndDate": "4712-12-31", "EffectiveStartDate": "2024-07-18", "LegalEntityId": "100000011593308", "PlanId": "300100182420223", "ValueCurrency": 500, "ValueFormat": "CURRENCY", "ValueString": "500", "Mode": "EDIT" }

▲ Initial Value

{ "LegalEntityId": "100000011593308", "PlanId": "300100182420223", "ComponentId": "300100182420225", "EffectiveStartDate": "2024-07-18", "EffectiveEndDate": "4712-12-31", "ValueFormat": "CURRENCY", "ValueCurrency": "45", "DisplayCurrency": "USD", "ValueString": "45", "Mode": "EDIT" }

ZCMP US VOL 401B Many

401B Many

07.18.2024 - Ongoing

Amount

USD 500.00

Processing Type

Recurring

⋮

▼

Show previous contributions

Cancel

Continue

Submit

ORACLE

9

