Oracle® E-Business Suite

Desktop Integration Framework Developer's Guide

Release 12.2

Part No. E22005-17

October 2023

Oracle E-Business Suite Desktop Integration Framework Developer's Guide, Release 12.2

Part No. E22005-17

Primary Author: Clara Jaeckel

Contributing Author: Padmaprabodh Ambale, Rekha Ayothi, Chen Jichao, Amanda Gonzalez, Hylvia Miro, Neelima Putrevu, Senthilkumar Ramalingam, Poorna Ramasamy, Ramkumar Sekar

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Contents

Send Us Your Comments

Preface

1 Introduction to Oracle E-Business Suite Desktop Integration Framework

2 Managing Integrators

3 Managing Components

4 Managing Parameters

A Loading Integrator Definitions

Index

Send Us Your Comments

Oracle E-Business Suite Desktop Integration Framework Developer's Guide, Release 12.2

Part No. E22005-17

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document. Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

Note: Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the new Oracle E-Business Suite Release Online Documentation CD available on My Oracle Support and www.oracle.com. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: appsdoc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at www.oracle.com.

Preface

Intended Audience

Welcome to Release 12.2 of the *Oracle E-Business Suite Desktop Integration Framework Developer's Guide.*

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

- The principles and customary practices of your business area.
- Computer desktop application usage and terminology.

If you have never used Oracle E-Business Suite, we suggest you attend one or more of the Oracle E-Business Suite training classes available through Oracle University.

See Related Information Sources on page viii for more Oracle E-Business Suite product information.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Structure

1 Introduction to Oracle E-Business Suite Desktop Integration Framework

2 Managing Integrators

3 Managing Components

4 Managing Parameters

A Loading Integrator Definitions

Related Information Sources

This book is included in the Oracle E-Business Suite Documentation Library. If this guide refers you to other Oracle E-Business Suite documentation, use only the latest Release 12.2 versions of those guides.

Online Documentation

All Oracle E-Business Suite documentation is available online (HTML or PDF).

- **Online Help** - Online help patches (HTML) are available on My Oracle Support.

- **Oracle E-Business Suite Documentation Library** - This library, which is included in the Oracle E-Business Suite software distribution, provides PDF documentation as of the time of each release.

- **Oracle E-Business Suite Documentation Web Library** - This library, available on the Oracle Help Center (https://docs.oracle.com/cd/E26401_01/index.htm), provides the latest updates to Oracle E-Business Suite Release 12.2 documentation. Most documents are available in PDF and HTML formats.

- **Release Notes** - For information about changes in this release, including new features, known issues, and other details, see the release notes for the relevant product, available on My Oracle Support.

- **Oracle Electronic Technical Reference Manual -** The Oracle Electronic Technical Reference Manual (eTRM) contains database diagrams and a detailed description of database tables, forms, reports, and programs for each Oracle E-Business Suite product. This information helps you convert data from your existing applications and integrate Oracle E-Business Suite data with non-Oracle applications, and write custom reports for Oracle E-Business Suite products. The Oracle eTRM is available as an application in Oracle E-Business Suite.

Related Guides

You should have the following related books on hand. Depending on the requirements of your particular installation, you may also need additional manuals or guides.

Oracle Application Framework Developer's Guide

This guide contains the coding standards followed by Oracle E-Business Suite Development to create applications with Oracle Application Framework. This guide is available in PDF format on My Oracle Support and as online documentation in JDeveloper 10g with Oracle Application Extension.

Oracle E-Business Suite Concepts

This book is intended for all those planning to deploy Oracle E-Business Suite Release 12.2, or contemplating significant changes to a configuration. After describing the Oracle E-Business Suite architecture and technology stack, it focuses on strategic topics, giving a broad outline of the actions needed to achieve a particular goal, plus any installation and configuration choices that are available.

Oracle E-Business Suite Electronic Technical Reference Manual User's Guide

This guide describes how to set up and navigate Oracle E-Business Suite Electronic Technical Reference Manual (eTRM) user interface in Oracle E-Business Suite. It also explains how to browse and search the Oracle eTRM repository to locate desired FND and database metadata and objects, and how to view object details, reports, and diagrams.

Oracle E-Business Suite Flexfields Guide

This guide provides flexfields planning, setup, and reference information for the Oracle E-Business Suite implementation team, as well as for users responsible for the ongoing maintenance of Oracle E-Business Suite product data. This guide also provides information on creating custom reports on flexfields data.

Oracle E-Business Suite Installation Guide: Using Rapid Install

This book describes how to run Rapid Install to perform a fresh installation of Oracle E-Business Suite Release 12.2 or to replace selected technology stack executables in an existing instance.

Oracle E-Business Suite Maintenance Guide

This guide explains how to patch an Oracle E-Business Suite system, describing the adop patching utility and providing guidelines and tips for performing typical patching operations. It also describes maintenance strategies and tools designed to help keep a system running smoothly.

Oracle E-Business Suite Multiple Organizations Implementation Guide

This guide describes the multiple organizations concept in Oracle E-Business Suite. It describes setting up and working effectively with multiple organizations in Oracle E-Business Suite.

Oracle E-Business Suite Security Guide

This guide contains information on a comprehensive range of security-related topics, including access control, user management, function security, data security, secure configuration, and auditing. It also describes how Oracle E-Business Suite can be integrated into a single sign-on environment.

Oracle E-Business Suite Setup Guide

This guide contains information on system configuration tasks that are carried out either after installation or whenever there is a significant change to the system. The activities described include defining concurrent programs and managers, enabling Oracle Applications Manager features, and setting up printers and online help.

Oracle E-Business Suite User's Guide

This guide explains how to navigate products, enter and query data, and run concurrent requests by means of the user interfaces (UI) of Oracle E-Business Suite. It includes basic information on setting preferences and customizing the UI. An introduction to Oracle Enterprise Command Centers is also included. Lastly, this guide describes accessibility features and keyboard shortcuts for Oracle E-Business Suite.

Oracle General Ledger User's Guide

This guide provides information on how to use Oracle General Ledger. Use this guide to learn how to create and maintain ledgers, ledger currencies, budgets, and journal entries. This guide also includes information about running financial reports.

Oracle Report Manager User's Guide

Oracle Report Manager is an online report distribution system that provides a secure and centralized location to produce and manage point-in-time reports. Oracle Report Manager users can be either report producers or report consumers. Use this guide for information on setting up and using Oracle Report Manager.

Oracle Web Applications Desktop Integrator Implementation and Administration Guide

Oracle Web Applications Desktop Integrator brings Oracle E-Business Suite functionality to a spreadsheet, where familiar data entry and modeling techniques can be used to complete Oracle E-Business Suite tasks. You can create formatted spreadsheets on your desktop that allow you to download, view, edit, and create Oracle E-Business Suite data, which you can then upload. This guide describes how to implement Oracle Web Applications Desktop Integrator and how to define mappings, layouts, style sheets, and other setup options.

Oracle XML Publisher Report Designer's Guide

Oracle XML Publisher is a template-based reporting solution that merges XML data with templates in RTF or PDF format to produce a variety of outputs to meet a variety of business needs. Using Microsoft Word or Adobe Acrobat as the design tool, you can create pixel-perfect reports from the Oracle E-Business Suite. Use this guide to design your report layouts.

This guide is available through the Oracle E-Business Suite online help. For more information, see: *Notes for Using Oracle Business Intelligence Publisher 10g in Oracle E-Business Suite Release 12.2*, My Oracle Support Knowledge Document 1640073.1.

Oracle XML Publisher Administration and Developer's Guide

Oracle XML Publisher is a template-based reporting solution that merges XML data with templates in RTF or PDF format to produce a variety of outputs to meet a variety of business needs. Outputs include: PDF, HTML, Excel, RTF, and eText (for EDI and EFT transactions). Oracle XML Publisher can be used to generate reports based on existing Oracle E-Business Suite report data, or you can use Oracle XML Publisher's data extraction engine to build your own queries. Oracle XML Publisher also provides a robust set of APIs to manage delivery of your reports via e-mail, fax, secure FTP, printer, WebDav, and more. This guide describes how to set up and administer Oracle

XML Publisher as well as how to use the Application Programming Interface to build custom solutions.

This guide is available through the Oracle E-Business Suite online help. For more information, see: *Notes for Using Oracle Business Intelligence Publisher 10g in Oracle E-Business Suite Release 12.2*, My Oracle Support Knowledge Document 1640073.1.

Integration Repository

The Oracle Integration Repository is a compilation of information about the service endpoints exposed by the Oracle E-Business Suite of applications. It provides a complete catalog of Oracle E-Business Suite's business service interfaces. The tool lets users easily discover and deploy the appropriate business service interface for integration with any system, application, or business partner.

The Oracle Integration Repository is shipped as part of the Oracle E-Business Suite. As your instance is patched, the repository is automatically updated with content appropriate for the precise revisions of interfaces in your environment.

Do Not Use Database Tools to Modify Oracle E-Business Suite Data

Oracle STRONGLY RECOMMENDS that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle E-Business Suite data unless otherwise instructed.

Oracle provides powerful tools you can use to create, store, change, retrieve, and maintain information in an Oracle database. But if you use Oracle tools such as SQL*Plus to modify Oracle E-Business Suite data, you risk destroying the integrity of your data and you lose the ability to audit changes to your data.

Because Oracle E-Business Suite tables are interrelated, any change you make using an Oracle E-Business Suite form can update many tables at once. But when you modify Oracle E-Business Suite data using anything other than Oracle E-Business Suite, you may change a row in one table without making corresponding changes in related tables. If your tables get out of synchronization with each other, you risk retrieving erroneous information and you risk unpredictable results throughout Oracle E-Business Suite.

When you use Oracle E-Business Suite to modify your data, Oracle E-Business Suite automatically checks that your changes are valid. Oracle E-Business Suite also keeps track of who changes information. If you enter information into database tables using database tools, you may store invalid information. You also lose the ability to track who has changed your information because SQL*Plus and other database tools do not keep a record of changes.

1

Introduction to Oracle E-Business Suite Desktop Integration Framework

This chapter covers the following topics:

- Overview

Overview

Oracle Web Applications Desktop Integrator brings Oracle E-Business Suite functionality to the desktop where the familiar Microsoft Excel, Word, and Project applications can be used to complete Oracle E-Business Suite tasks. Oracle Web Applications Desktop Integrator uses a set of metadata called an integrator to encapsulate all the information needed to integrate a particular Oracle E-Business Suite task with a desktop application. Several Oracle E-Business Suite applications provide seeded integrators for their functionality. With Oracle E-Business Suite Desktop Integration Framework, you can define custom integrators for additional tasks according to your users' needs.

- Tasks in a standard Oracle E-Business Suite application that are not covered by seeded integrators
- Tasks in custom applications developed for your instance

By defining an integrator, you enable your users to do the following.

- Create a formatted document such as a spreadsheet on the desktop containing application-specific fields for data entry.
- Import data into the desktop document from the database or a text file.
- Work with the data in the desktop application while enforcing the owning application's business rules.
- Upload data from the desktop document into Oracle E-Business Suite.

- Validate the data being uploaded and receive immediate feedback about the results of the validation during the upload process.

Oracle E-Business Suite Desktop Integration Framework provides a graphical user interface in which developers can define integrators and all associated supporting objects, without needing to work directly with the underlying Oracle Web Applications Desktop Integrator tables and APIs. Through this user interface, Oracle E-Business Suite Desktop Integration Framework reduces development time, increases developer productivity, and enhances ease of maintenance for the integrators you define.

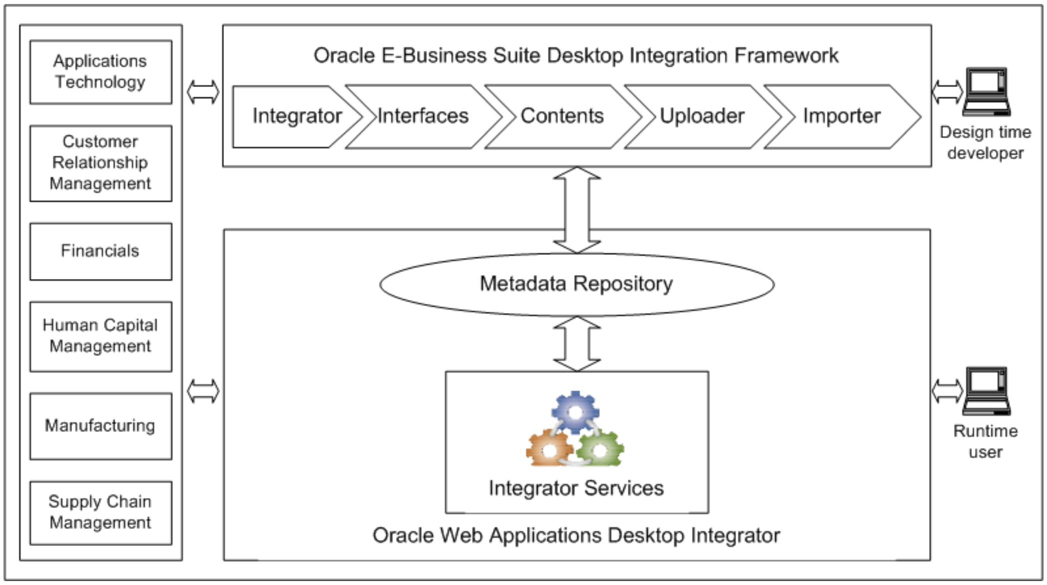
Oracle E-Business Suite Desktop Integration Framework Architecture

As shown in the following diagram, Oracle E-Business Suite product families including Applications Technology, Customer Relationship Management, Financials, Human Capital Management, Manufacturing, and Supply Chain Management leverage Oracle E-Business Suite Desktop Integration Framework and Oracle Web Applications Desktop Integrator.

The Oracle E-Business Suite Desktop Integration Framework user interface lets a design-time developer define all the metadata required to create a custom integrator, accessing design information from the associated Oracle E-Business Suite application as needed. The integrator metadata includes basic integrator properties, interfaces, contents, an uploader, and an importer. The integrator metadata is stored in the Oracle Web Applications Desktop Integrator metadata repository, alongside the metadata for seeded integrators.

When a runtime user downloads or uploads data between Oracle E-Business Suite and a desktop document, the runtime integrator services for Oracle Web Applications Desktop Integrator use the metadata stored in the repository to perform the specified task in the associated Oracle E-Business Suite application.

Oracle E-Business Suite Desktop Integration Framework Architecture



Process Flow for Integrator Definition

An *integrator* is a set of metadata that encapsulates all the information needed to integrate a particular Oracle E-Business Suite task with a desktop application.

- You can define integrators that only download data from Oracle E-Business Suite into a desktop document and do not upload data. The document serves as a report of the Oracle E-Business Suite data that can be viewed in the desktop application. This type of integrator is called a *reporting-only integrator*.

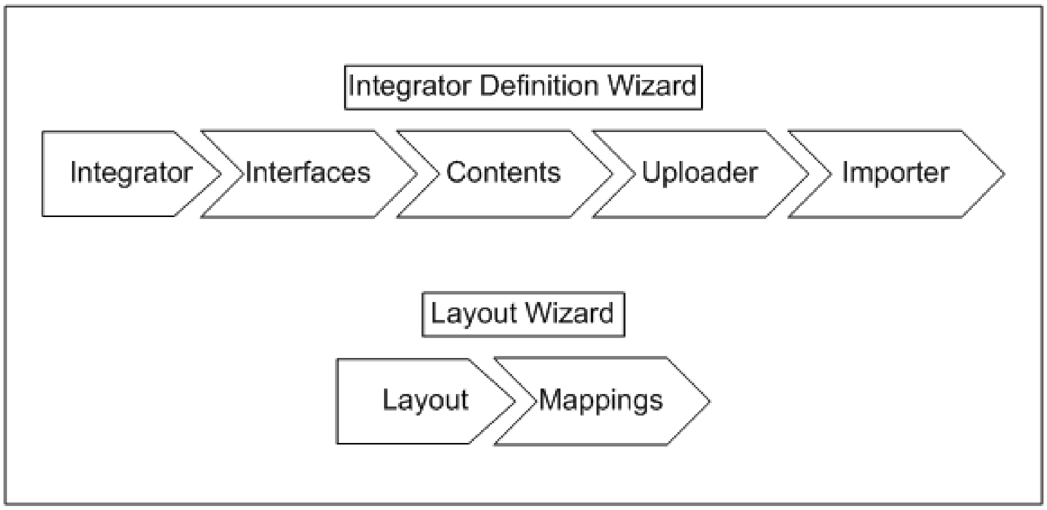
- You can also define integrators that create a desktop document and then upload data from that document into Oracle E-Business Suite. An integrator can create a document in one of these ways.

  - Create a blank document that is laid out and formatted according to an Oracle E-Business Suite application's business rules but contains no data. Users must enter all data into the document manually.

  - Create a document that is initially populated with data from a text file. Users can work with the data in the desktop application before uploading it to Oracle E-Business Suite.

  - Create a document that is initially populated with data downloaded from Oracle E-Business Suite through a SQL query or Java class. Users can work with the data in the desktop application before uploading it back to Oracle E-Business Suite.

Process Flow for Integrator Definition



As shown in the process flow diagram, Oracle E-Business Suite Desktop Integration Framework provides a wizard interface to guide you through the main steps for defining integrators. In the integrator definition wizard, you can perform the following tasks:

- Define basic integrator properties.
- Define interfaces for the integrator. An *interface* is a set of metadata that determines how the integrator uploads data from a desktop document to Oracle E-Business Suite. An *interface attribute* defines a single data column within the interface.

  You do not need to define an interface for a reporting-only integrator.
- Define contents for the integrator. A *content* is a set of metadata that determines what data the integrator uses to populate the desktop document that it creates. A *content column* defines a single data column within the content. You can define these content types:
  - Text File - Populate the desktop document with data from a text file.
  - SQL Query - Populate the desktop document with data from a SQL query that selects from Oracle E-Business Suite application tables.
  - Java - Populate the desktop document with data returned by a Java class from Oracle E-Business Suite application tables.

  You should always define at least one SQL query or Java content for a reporting-only integrator. If you want the integrator to create a blank document initially, then you do not need to define a content for the integrator.
- Define an uploader for the integrator. An *uploader* is a set of metadata that

determines what options are available to a user in the upload parameters window when the user uploads data from a desktop document to Oracle E-Business Suite.

You do not need to define an uploader for a reporting-only integrator.

- Define an importer for the integrator. An *importer* is a set of metadata that determines how Oracle Web Applications Desktop Integrator moves uploaded data from Oracle E-Business Suite interface tables to base application tables. Define an importer if you want to invoke an import process automatically after uploading data from the desktop document to the interface tables. An importer can invoke one of these types of import processes:

    - Asynchronous concurrent request

    - Synchronous concurrent request

    - PL/SQL API

  You do not need to define an importer for a reporting-only integrator.

Oracle E-Business Suite Desktop Integration Framework also provides a user interface that lets you manage and create components for use in integrators. A *component* is a set of metadata that provides a list of values for a field in a document.

Additionally, Oracle E-Business Suite Desktop Integration Framework provides a user interface that lets you define parameters for an integrator.

> **Note:** To ensure that your customizations are handled appropriately in conjunction with the Online Patching feature introduced in Release 12.2, follow the instructions in *Deploying Customizations in Oracle E-Business Suite Release 12.2*, My Oracle Support Document 1577661.1.

After completing the main integrator definition in the wizard, you can also optionally define layouts and mappings for use with your integrator. A *layout* is a set of metadata that determines what fields appear in the desktop document and how those fields are displayed and formatted. A *mapping* is a set of metadata that links content columns to the corresponding interface attributes. You can create layouts and mappings for your users to use with the integrator you defined. Depending on the access granted to them, users can also define layouts and mappings themselves. See: *Oracle Web Applications Desktop Integrator Implementation and Administration Guide*.

2

Managing Integrators

This chapter covers the following topics:

- Managing Integrators
- Defining Integrators

Managing Integrators

An *integrator* is a set of metadata that encapsulates all the information needed to integrate a particular Oracle E-Business Suite task with a desktop application. Use the Integrators page to view and manage the integrators defined for your Oracle E-Business Suite installation.

To manage integrators:

1. Navigate to the Integrators page from the Oracle E-Business Suite Navigator by choosing either Desktop Integration Manager: Manage Integrators or a custom responsibility and menu path specified by your system administrator.

2. Search for the integrators you want to display. You can enter the following search criteria:

   - Integrator Name - Enter the display name of the integrator. You can enter a partial value to search for integrators whose names contain that value.

   - Internal Name - Enter the unique internal name of the integrator. You can enter a partial value to search for integrators whose names contain that value.

   - Application - Select the application that owns the integrator.

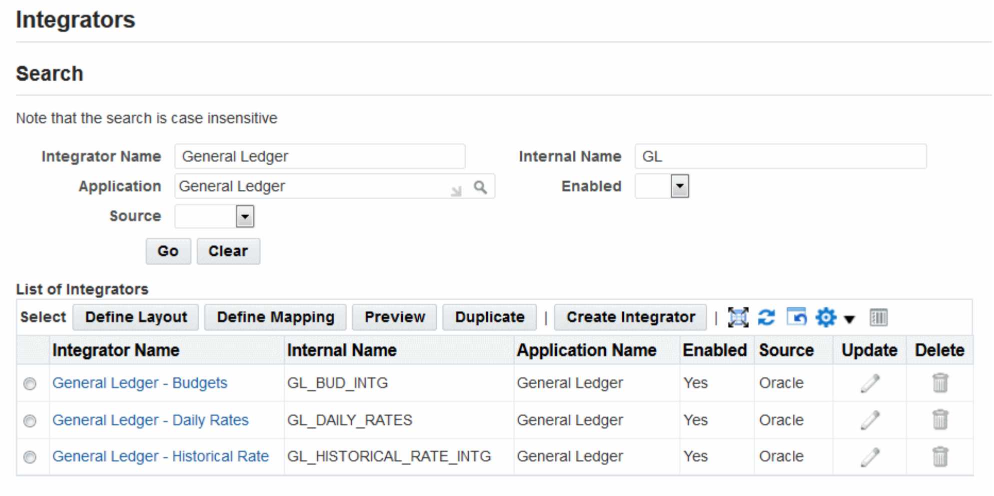
   - Enabled - Select Yes to search for integrators that are enabled or No to search for integrators that are disabled.

   - Source - Select Oracle to search for integrators seeded by Oracle E-Business

Suite or Custom to search for custom integrators.

The search criteria fields are all case-insensitive.

Integrators Page



Integrators

Search

Note that the search is case insensitive

Integrator Name	General Ledger	Internal Name	GL
Application	General Ledger	Enabled	
Source			

List of Integrators

	Integrator Name	Internal Name	Application Name	Enabled	Source	Update	Delete
○	General Ledger - Budgets	GL_BUD_INTG	General Ledger	Yes	Oracle	✎	🗑
○	General Ledger - Daily Rates	GL_DAILY_RATES	General Ledger	Yes	Oracle	✎	🗑
○	General Ledger - Historical Rate	GL_HISTORICAL_RATE_INTG	General Ledger	Yes	Oracle	✎	🗑

3. To view an integrator definition in read-only mode, choose the integrator name link for that integrator. See: Defining Integrators, page 2-3.

4. To update an integrator, choose the update icon for that integrator. See: Defining Integrators, page 2-3.

   Note: You can only update custom integrators.

5. To delete an integrator, choose the delete icon for that integrator.

   Note: You can only delete custom integrators.

6. To define a layout for an integrator, select the integrator and select the Define Layout button. See: Defining Layouts, *Oracle Web Applications Desktop Integrator Implementation and Administration Guide*.

7. To define a mapping for an integrator, select the integrator and select the Define Mapping button. See: Defining Mappings, *Oracle Web Applications Desktop Integrator Implementation and Administration Guide*.

8. To test how an integrator creates a desktop document, select the integrator and select the Preview button. Oracle Web Applications Desktop Integrator then displays the Create Document interface, with the specified integrator selected, to let

you finish creating the preview document. See: Creating a Spreadsheet, *Oracle Web Applications Desktop Integrator Implementation and Administration Guide*.

> **Note:** When you access the Create Document page in preview mode, the resulting document is intended only for testing purposes to help you check the integrator definition. Consequently, the Save button for saving a shortcut and the Clear button are not displayed.

9. To define a new integrator, select the Create Integrator button. See: Defining Integrators, page 2-3.

10. To define a new integrator by copying an existing integrator, select the existing integrator and select the Duplicate button. See: Defining Integrators, page 2-3.

    > **Note:** Some seeded integrators contain product-specific features that cannot be defined or updated through the integrator definition wizard. These integrators cannot be duplicated.

Defining Integrators

Use the integrator definition wizard to create a new integrator, create an integrator by duplicating an existing integrator, update an existing integrator, or view the definition for an existing integrator.

> **Note:** You can only update custom integrators.

To define an integrator:

1. Navigate to the integrator definition wizard by one of the following methods:

    - Choose the Create Integrator button in the Integrators page.

    - Choose the Duplicate button in the Integrators page.

    - Choose the update icon in the Integrators page.

    - From the Oracle E-Business Suite Navigator, choose either Desktop Integration Manager: Create Integrator or a custom responsibility and menu path specified by your system administrator.

    If you are duplicating or updating an existing integrator, then the fields in the wizard are populated with previously defined information for the selected integrator.

2. In the first step of the wizard, define the basic properties of the integrator. See: To define integrator information, page 2-5.

3. If you want to use this integrator to upload data from a desktop document to Oracle E-Business Suite, define an interface for the integrator in the second step of the wizard. See: To define interfaces, page 2-7.

4. In the third step of the wizard, optionally define contents for the integrator. See: To define contents, page 2-27.

5. If you want to use this integrator to upload data from a desktop document to Oracle E-Business Suite, define the uploader for the integrator in the fourth step of the wizard. See: To define an uploader, page 2-32.

6. If you want to use this integrator to import uploaded data from Oracle E-Business Suite interface tables to application base tables, define the importer for the integrator in the fifth step of the wizard. See: To define an importer, page 2-38.

To view an integrator definition:

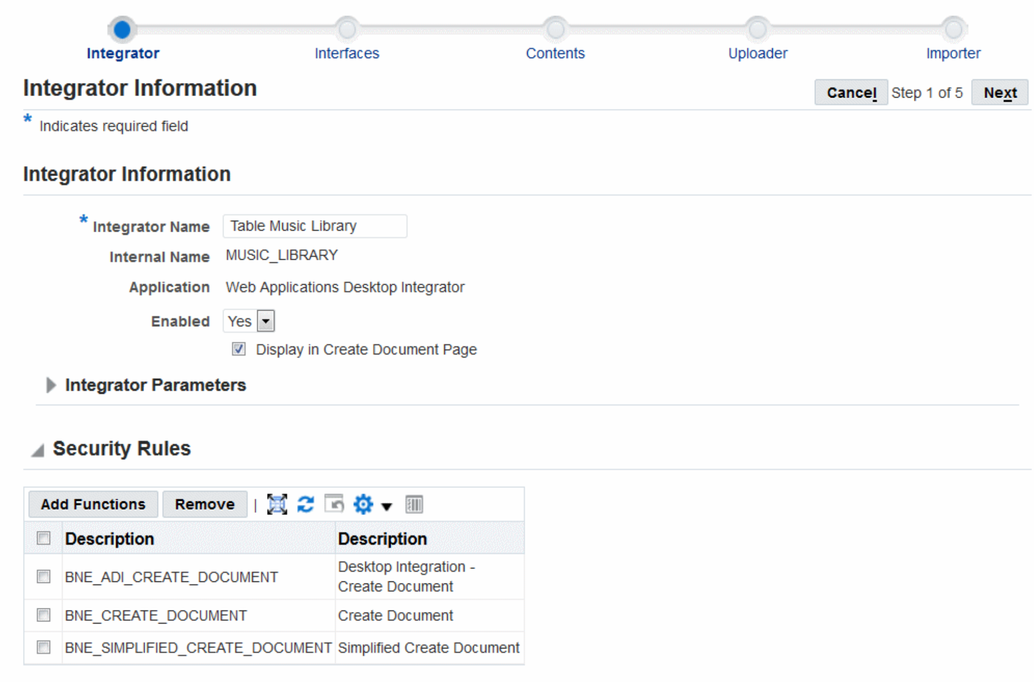
1. Navigate to the integrator definition wizard by choosing the integrator name link in the Integrators page. The integrator definition appears in the wizard in read-only mode.

2. Use the Next and Back buttons to navigate among the pages of the wizard. For more information about the fields displayed in each page, see: To define integrator information, page 2-5, To define interfaces, page 2-7, To define contents, page 2-27, To define an uploader, page 2-32, and To define an importer, page 2-38.

3. Choose the Cancel button in the Integrator Information, Interfaces, Contents, Uploader, or Importer page to return to the Integrators page.

To define integrator information:

Integrator Information Page



Integrator Information

* Indicates required field

Integrator Information

* Integrator Name	Table Music Library
Internal Name	MUSIC_LIBRARY
Application	Web Applications Desktop Integrator
Enabled	Yes
	☑ Display in Create Document Page

Integrator Parameters

Security Rules

□	Description	Description
□	BNE_ADI_CREATE_DOCUMENT	Desktop Integration - Create Document
□	BNE_CREATE_DOCUMENT	Create Document
□	BNE_SIMPLIFIED_CREATE_DOCUMENT	Simplified Create Document

1. In the Integrator Information region of the Integrator Information page, enter the display name of the integrator.

   When you duplicate an integrator, Oracle E-Business Suite Desktop Integration Framework creates a default display name for the new integrator by adding `Copy of` before the name of the original integrator. It is recommended that you update the display name to better describe the new integrator.

2. Enter a unique internal name for the integrator. The internal name can include only the following characters:

   - Uppercase letters from `A` through `Z`
   - Numerals from `0` through `9`
   - The underscore character (_)

   Oracle E-Business Suite Desktop Integration Framework uses the internal name of the integrator to generate internal names for the supporting objects associated with the integrator, such as interfaces, contents, layouts, and mappings.

   When you duplicate an integrator, Oracle E-Business Suite Desktop Integration Framework creates a default internal name for the new integrator by adding `COPY_` before the name of the original integrator. It is recommended that you update the internal name to better describe the new integrator.

> **Note:** To preserve consistency, you cannot update the internal name of an existing integrator.

3. Select the application that owns the integrator.

   > **Note:** To preserve consistency, you cannot update the application of an existing integrator. You also cannot update the application for an integrator created as a duplicate of an existing integrator.

4. If you want to use this integrator only for exporting data from Oracle E-Business Suite to a desktop document, select the Reporting Only option. In this case the document serves as a report of the Oracle E-Business Suite data that can be viewed in the desktop application. If you want to use this integrator to upload data from a desktop document to Oracle E-Business Suite, do not select this option.

5. In the Enabled field, select Yes to enable this integrator or No to disable this integrator. If you disable an integrator, its definition remains available for reference, but you cannot use it to define layouts or mappings, create a document, or download or upload Oracle E-Business Suite data.

6. If you want your users to be able to select this integrator in the Create Document interface within Oracle Web Applications Desktop Integrator, select the Display in Create Document Page option. In this case you must also add the relevant functions as security rules for this integrator in step 11.

   If you will include the functionality to create documents within the functional page flow of your own application, then you do not need to select this option.

7. To specify parameters for the integrator, expand the Integrator Parameters region. See: Managing Parameters, page 4-1.

8. Select the server-side upload parameter list.

9. Select the document parameter list that contains any runtime parameters that can be referenced in layouts for the integrator to be included when a document is created. See: To define a document parameter list, page 4-17 and Defining Layouts, *Oracle Web Applications Desktop Integrator Implementation and Administration Guide*.

10. Select the session parameter list.

11. The Security Rules region displays a list of functions through which the integrator can be accessed. A user must have menu access to at least one of these functions in

order to use this integrator, including creating a desktop document, downloading data into the document, using a list of values based on Oracle E-Business Suite data within the desktop document, and uploading data to Oracle E-Business Suite. To add functions to the list, select Add Functions. You can repeat this step to allow access through additional functions.

If you selected the Display in Create Document Page option in step 6, then you must add the following two functions as security rules to enable this integrator to appear in the list of integrators that users can select in the Create Document interface:

- `BNE_CREATE_DOCUMENT`
- `BNE_ADI_CREATE_DOCUMENT`

12. Select Next.

    - If you did not select the Reporting Only option, then the integrator definition wizard displays the Interfaces page. See: To define interfaces, page 2-7.
    - If you selected the Reporting Only option, then you do not need to define an interface. In this case the integrator definition wizard proceeds directly to the Contents page. See: To define contents, page 2-27.

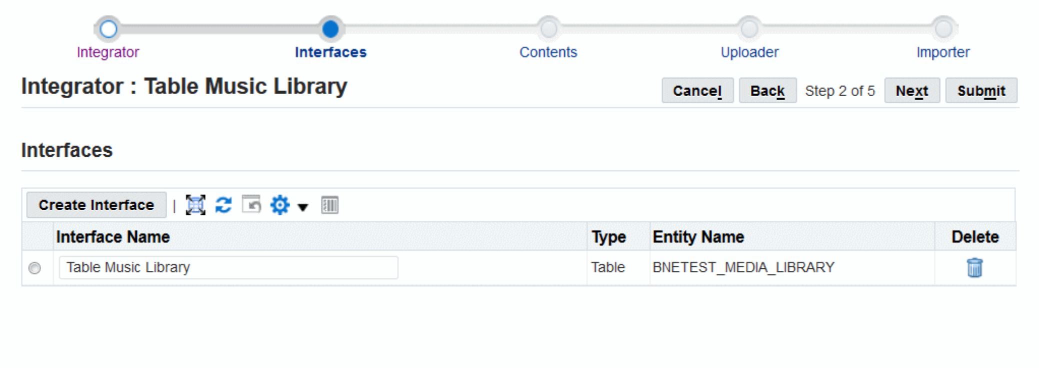
To define interfaces:

An *interface* is a set of metadata that determines how the integrator uploads data from a desktop document to Oracle E-Business Suite. Use the Interfaces page of the integrator definition wizard to create and manage an interface.

> **Important:** Oracle Web Applications Desktop Integrator currently supports only one interface per integrator. Consequently, when you define a custom integrator, you should create only one interface in the Interfaces page.

> **Note:** If you selected the Reporting Only option while defining the integrator information, do not manually define an interface for the integrator. Oracle E-Business Suite Desktop Integration Framework automatically defines a default placeholder interface for reporting-only integrators.

Interfaces Page



Integrator : Table Music Library

Interfaces

	Interface Name	Type	Entity Name	Delete
○	Table Music Library	Table	BNETEST_MEDIA_LIBRARY	🗑

1. If an interface is already defined for this integrator, the Interfaces page displays the list of interfaces.

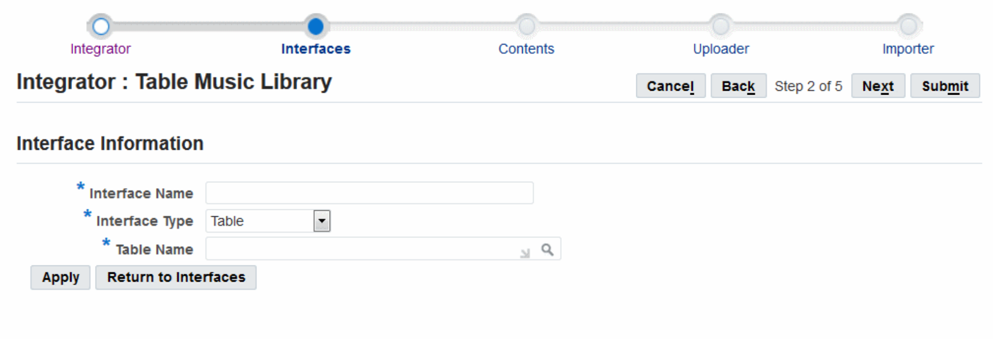
   - To update the name of an interface, enter the new name in the Interface Name field.

   - To delete an interface, choose the delete icon for that interface.

     > **Note:** When you duplicate an integrator, you cannot delete an interface that was copied from the original integrator.

   - To create a new interface, select the Create Interface button. The Interface Information region appears.

   Interface Information Region

   Integrator : Table Music Library

   Interface Information

   * Interface Name

   * Interface Type: Table

   * Table Name

   If no interface has yet been defined for the integrator, then the Interfaces page displays the Interface Information region in place of the interfaces list.

2. In the Interface Information region, enter the interface name.

3. Select the interface type, either table, API - procedure, or API - function.

   > **Note:** When you duplicate an integrator, you cannot update the interface type for an interface that was copied from the original integrator.

4. For a table interface, select the table name. Oracle E-Business Suite Desktop Integration Framework creates an interface attribute for each column in the table.

   If you create custom application tables, you must register those tables to make them available for selection in the Table Name field. See: Table Registration API, *Oracle E-Business Suite Developer's Guide*.

   > **Note:** The values available for selection in the Table Name field include only tables. You cannot create a table interface based on a view.

   > **Note:** When you duplicate an integrator, you cannot update the table name for an interface that was copied from the original integrator.

5. For a procedure interface, enter the package name and procedure name. Oracle E-Business Suite Desktop Integration Framework creates an interface attribute for each input argument in the procedure. You can also optionally specify whether the procedure returns an error message or an FND message code if it encounters an error.

   > **Note:** The PL/SQL procedure for an interface must not contain any commit statements. A commit statement in this procedure can cause inconsistencies during data upload: if an upload is submitted with both valid and invalid records, the valid records may be uploaded although Oracle Web Applications Desktop Integrator returns a message that no rows were uploaded.

   > **Note:** When you duplicate an integrator, you cannot update the package name and procedure name for an interface that was copied from the original integrator.

6. For a function interface, enter the package name and function name. Oracle E-Business Suite Desktop Integration Framework creates an interface attribute for each input argument in the function. If the function returns more details about any error that it encounters, select the Returns Error Information option. You can also

optionally specify whether the function returns an error message or an FND message code if it encounters an error.

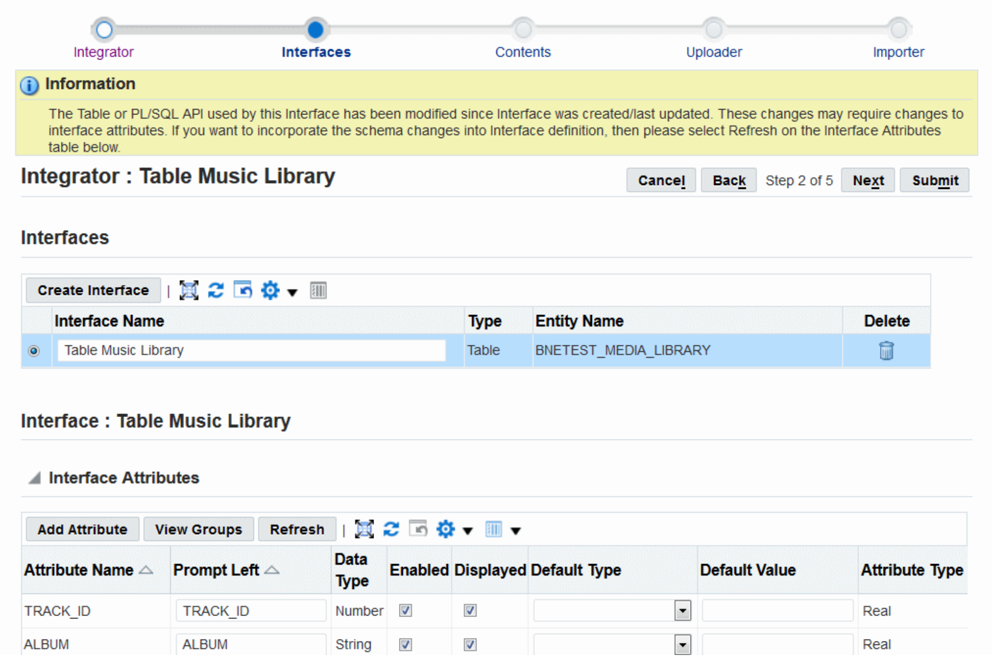
> **Note:** The PL/SQL function for an interface must not contain any commit statements. A commit statement in this function can cause inconsistencies during data upload: if an upload is submitted with both valid and invalid records, the valid records may be uploaded although Oracle Web Applications Desktop Integrator returns a message that no rows were uploaded.

> **Note:** When you duplicate an integrator, you cannot update the package name and function name for an interface that was copied from the original integrator.

7. Choose Apply.

8. In the Interfaces page, select an interface in the list to display its details, including interface attributes.

Interfaces Page with Interface Attributes Region and Refresh Message



Information

The Table or PL/SQL API used by this Interface has been modified since Interface was created/last updated. These changes may require changes to interface attributes. If you want to incorporate the schema changes into Interface definition, then please select Refresh on the Interface Attributes table below.

Integrator : Table Music Library

Interfaces

	Interface Name	Type	Entity Name	Delete
●	Table Music Library	Table	BNETEST_MEDIA_LIBRARY	🗑

Interface : Table Music Library

Interface Attributes

Attribute Name	Prompt Left	Data Type	Enabled	Displayed	Default Type	Default Value	Attribute Type
TRACK_ID	TRACK_ID	Number	☑	☑			Real
ALBUM	ALBUM	String	☑	☑			Real

- For a procedure interface, you can optionally update whether the procedure returns an error message or an FND message code if it encounters an error.

- For a function interface, you can optionally update whether the function returns more details about any error that it encounters and whether the function returns an error message or an FND message code if it encounters an error.

9. The Interface Attributes region initially displays the attributes that were automatically created for the interface based on the specified table columns or API arguments. These attributes are referred to as *real* interface attributes. To define an additional *custom* interface attribute that can appear in a layout, choose Add Attribute. See: To define an interface attribute, page 2-16.

   Note: If you want to define an attribute group, you should define a custom attribute to use as the parent attribute for the group.

10. You can optionally choose Refresh in the Interface Attributes region to update the real interface attributes if the definition of the underlying table, procedure, or function has changed. The Interfaces page displays a message to notify you if Oracle E-Business Suite Desktop Integration Framework detects that the underlying object definition has changed since the interface was last updated.

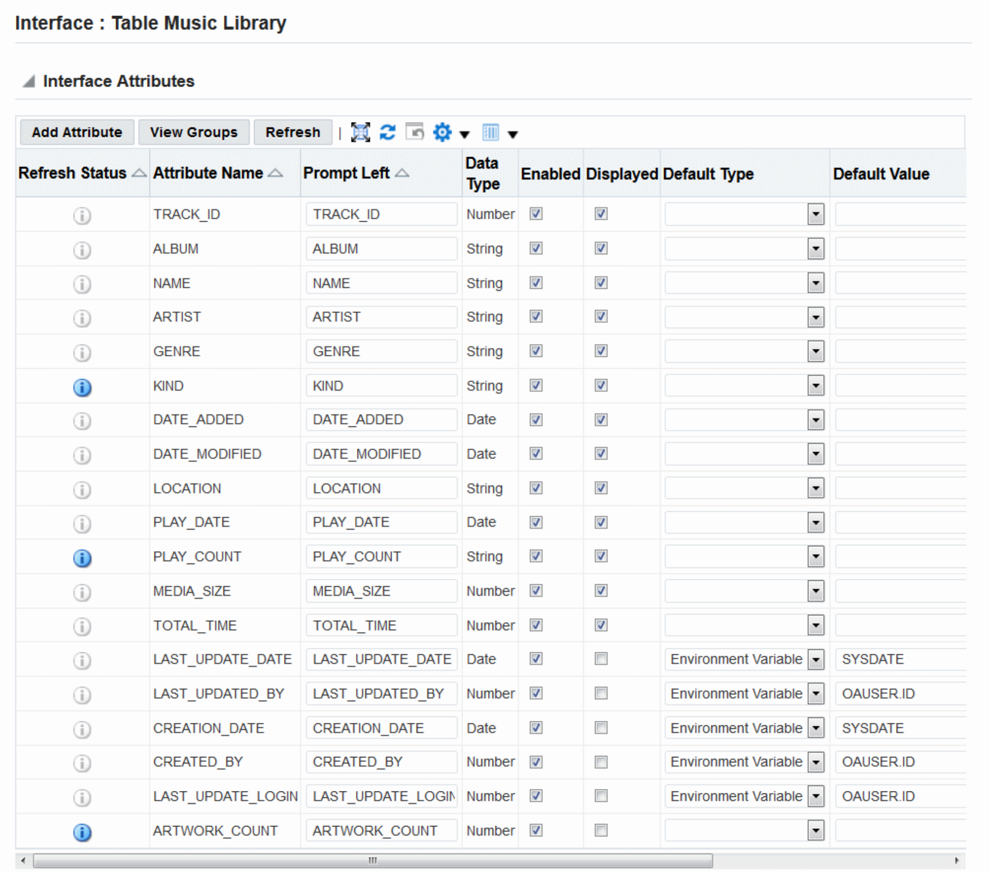
    Oracle E-Business Suite Desktop Integration Framework can automatically refresh the interface and its real interface attributes for the following types of changes in the underlying object definition:

- For new columns added at the end of the underlying table, or new input arguments added at the end of the argument list for a procedure or function, Oracle E-Business Suite Desktop Integration Framework adds the corresponding new real interface attributes.

- For changes in the data type of a table column or API argument, the size of a table column, or whether a table column can be null, Oracle E-Business Suite Desktop Integration Framework updates the corresponding interface attribute details.

- If a function now returns error information when it previously did not, or vice versa, Oracle E-Business Suite Desktop Integration Framework updates the corresponding detail in the interface definition.

Interfaces Page - Interface Attributes Region with Refresh Status Column



Interface : Table Music Library

Interface Attributes

Refresh Status	Attribute Name	Prompt Left	Data Type	Enabled	Displayed	Default Type	Default Value
ⓘ	TRACK_ID	TRACK_ID	Number	☑	☑		
ⓘ	ALBUM	ALBUM	String	☑	☑		
ⓘ	NAME	NAME	String	☑	☑		
ⓘ	ARTIST	ARTIST	String	☑	☑		
ⓘ	GENRE	GENRE	String	☑	☑		
ℹ	KIND	KIND	String	☑	☑		
ⓘ	DATE_ADDED	DATE_ADDED	Date	☑	☑		
ⓘ	DATE_MODIFIED	DATE_MODIFIED	Date	☑	☑		
ⓘ	LOCATION	LOCATION	String	☑	☑		
ⓘ	PLAY_DATE	PLAY_DATE	Date	☑	☑		
ℹ	PLAY_COUNT	PLAY_COUNT	String	☑	☑		
ⓘ	MEDIA_SIZE	MEDIA_SIZE	Number	☑	☑		
ⓘ	TOTAL_TIME	TOTAL_TIME	Number	☑	☑		
ⓘ	LAST_UPDATE_DATE	LAST_UPDATE_DATE	Date	☑	□	Environment Variable	SYSDATE
ⓘ	LAST_UPDATED_BY	LAST_UPDATED_BY	Number	☑	□	Environment Variable	OAUSER.ID
ⓘ	CREATION_DATE	CREATION_DATE	Date	☑	□	Environment Variable	SYSDATE
ⓘ	CREATED_BY	CREATED_BY	Number	☑	□	Environment Variable	OAUSER.ID
ⓘ	LAST_UPDATE_LOGIN	LAST_UPDATE_LOGIN	Number	☑	□	Environment Variable	OAUSER.ID
ℹ	ARTWORK_COUNT	ARTWORK_COUNT	Number	☑	□		

After you choose Refresh, the Interface Attributes region displays a Refresh Status column with an icon for each attribute. The icon is enabled if the underlying table column or API argument for the attribute has changed and disabled if there was no change. You can sort the list by the Refresh Status column to group all the changed attributes at the beginning or end of the list for easier review. For the changed attributes, choose the icon in the Refresh Status column to open a popup window that displays the details of the changes. To save the changes and complete the refresh, choose Submit.

After you refresh interface attributes for an existing integrator that allows uploading data to Oracle E-Business Suite, you should also update the content, mapping, and layout for the integrator as needed to accommodate the changed interface attributes.

> **Note:** You do not need to perform a refresh for the default placeholder interface for a reporting-only integrator. If you add content columns for a reporting-only integrator, Oracle E-Business

Suite Desktop Integration Framework automatically updates the default placeholder interface and mapping lines accordingly.

Note: Oracle E-Business Suite Desktop Integration Framework cannot automatically refresh interface attributes for certain types of changes in the underlying object definition, including the following

- Columns were removed from the underlying table, or input arguments were removed from the underlying procedure or function.
- Columns or arguments were added in a position other than at the end of the existing sequence.
- The name of a column or argument was changed.
- An underlying procedure was changed to a function, or vice versa.

If the updated table or API definition includes any of these changes, then Oracle E-Business Suite Desktop Integration Framework displays an error message when you attempt to perform a refresh. In this case no refresh is performed and no changes are made in the interface definition. Instead, you must manually create a new interface definition to specify how the integrator should work with the updated table or API.

11. You can update some attribute details directly in the Interface Attributes region.

    - Enter a prompt to appear to the left of the attribute if it appears in the header region of a document. This prompt is also used as the display name of the attribute.
    - Select whether the attribute is enabled for use in a layout.
    - Select whether the attribute is available for display in a layout.
    - Optionally select the source type from which the default value for the attribute is derived, either Constant, Environment Variable, Parameter, Profile Option, Sequence, SQL Query, or Lookup.
    - Enter a default value for the attribute corresponding to the selected type.

      Note: For an attribute that contains a date value, the format of the default value should match the date format defined for the integrator in the integrator LDT file.

You can also choose the update icon for an attribute to update additional attribute details. See: To update an interface attribute, page 2-20.

12. To delete a custom interface attribute, choose the delete icon for that attribute.

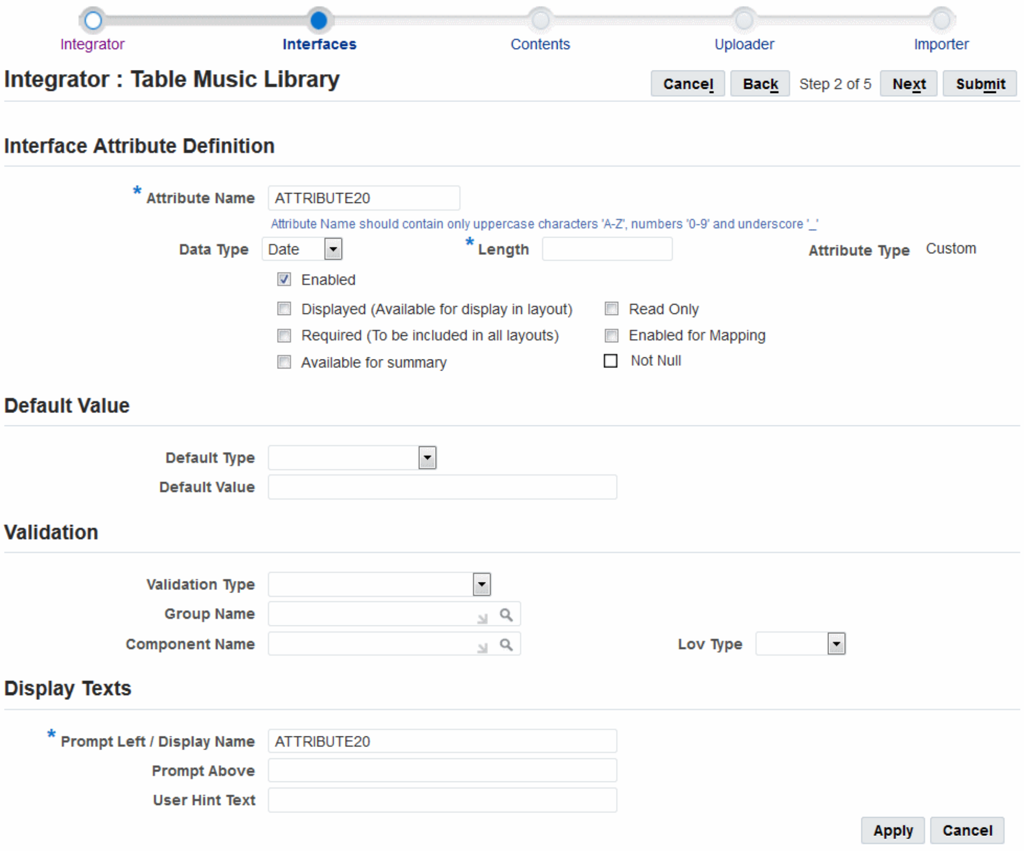
    Note: You cannot delete real interface attributes.

13. If validation of one interface attribute depends on the value of another attribute, or if several attributes are validated the same way, you can associate those attributes with each other by defining an attribute group and perform validation for all attributes in the group together. To view and manage groups, choose View Groups. See: To manage interface attribute groups, page 2-24.

14. To proceed to the Contents page, select Next, or to save this integrator definition without defining contents or an uploader or importer, select Submit.

To define an interface attribute:

Interface Attribute Definition Page



Integrator : Table Music Library

Interface Attribute Definition

* Attribute Name: ATTRIBUTE20

Attribute Name should contain only uppercase characters 'A-Z', numbers '0-9' and underscore '_'

Data Type: Date | * Length: | Attribute Type: Custom

☑ Enabled

☐ Displayed (Available for display in layout) | ☐ Read Only

☐ Required (To be included in all layouts) | ☐ Enabled for Mapping

☐ Available for summary | ☐ Not Null

Default Value

Default Type:

Default Value:

Validation

Validation Type:

Group Name:

Component Name: | Lov Type:

Display Texts

* Prompt Left / Display Name: ATTRIBUTE20

Prompt Above:

User Hint Text:

1. In the Interface Attribute Definition page, enter a unique internal name for the attribute. The name can include only the following characters:
    - Uppercase letters from A through Z
    - Numerals from `0` through `9`
    - The underscore character (_)

2. Select the attribute data type.
    - Number - This data type holds a number value.
    - String - This data type holds a string value.
    - Date - This data type holds a date value without a time component.

- DateTime - This data type holds a date value with a time component that is both stored in the Oracle E-Business Suite database and displayed in the desktop document according to the server time zone.
- DateTimeTZ - This data type holds a date value with a time component that is stored in the Oracle E-Business Suite database according to the server time zone but is displayed in the desktop document according to the user's preferred time zone, as specified in the Client Timezone (CLIENT_TIMEZONE_ID) profile option. Oracle Web Applications Desktop Integrator converts values with this data type to the client time zone when they are downloaded to a document and converts these values to the server time zone when they are uploaded to Oracle E-Business Suite.

  > **Additional Information:** See: User-Preferred Time Zones, *Oracle E-Business Suite Setup Guide*.

3. For an attribute derived from an interface table column with a data type of number or string, review the attribute length, which is determined by the size of the corresponding interface table column.
4. Select whether the attribute is enabled for use in a layout.
5. Select whether the attribute is available for display in a layout.
6. Select whether the attribute is required to be included in all layouts for this integrator.
7. Select whether the attribute can be summed to form a total value.
8. Select whether the attribute is read-only. A read-only attribute is placed in the context section of the desktop document and users cannot change its value.
9. Select whether the attribute is enabled for use in a mapping, to be mapped to a content column.
10. For an attribute derived from an interface table column, the Not Null checkbox indicates whether that interface table column must not be null.
11. Optionally select the source type from which the default value for the attribute is derived, either Constant, Environment Variable, Parameter, Profile Option, Sequence, SQL Query, or Lookup.

    > **Note:** The default value for an attribute is not inserted into the document when the document is created. Instead, if no other value has been specified for the attribute, the default value is derived and

inserted when a user performs an upload.

If you select Sequence as the source type, then you should also mark this attribute as read-only, and you should specify the name of a database sequence as the default value. In this case, when a user performs an upload, the upload process sets the attribute value for each row that is uploaded to the next value generated by that database sequence (`Sequence.NextValue`). That is, each row will be assigned a separate sequence value.

You can also define an interface attribute that assigns the same sequence value to all rows processed in a particular upload, to serve as an upload batch ID. To define this type of attribute, select SQL Query as the source type, mark this attribute as read-only, and set the default value to the following query:

```
Select SequenceName.nextval from dual
```

12. Enter a default value for the attribute corresponding to the selected type.

    Note: For an attribute that contains a date value, the format of the default value should match the date format defined for the integrator in the integrator LDT file.

13. Select the validation type for the attribute, either Descriptive Flexfield, Descriptive Flexfield Context, Descriptive Flexfield Segment, Group, Java, Key Flexfield, Key Flexfield ID, Key Flexfield ID Segment, Key Flexfield Segment, or Table. If you only want to validate the data type, length, and not null requirement for the attribute, then you can leave the validation type blank.

14. If the attribute should be validated as part of an attribute group, select the group name. See: To manage interface attribute groups, page 2-24.

15. For descriptive flexfield validation, specify the following validation details.

    - Enter the validation entity for the attribute. The default validation entity is `oracle.apps.bne.integrator.validators.BneDFFValidator`.

    - Select the descriptive flexfield to use for validation.

    - Select Yes or No to indicate whether the descriptive flexfield segment values should be concatenated.

16. For descriptive flexfield context validation, specify the following validation details.

    - Select the descriptive flexfield to use for validation.

- Select Yes or No to indicate whether the descriptive flexfield segment values should be concatenated.

17. For descriptive flexfield segment validation, specify the following validation details.

    - Enter the segment number.

18. For group validation or Java validation, specify the following validation details.

    - Enter the validation entity for the attribute.

19. For key flexfield or key flexfield ID validation, specify the following validation details.

    - Enter the validation entity for the attribute. The default validation entity is `oracle.apps.bne.integrator.validators.BneKFFValidator`.

    - Select the key flexfield to use for validation.

    - Select Yes or No to indicate whether the key flexfield segment values should be concatenated.

    > **Note:** To use a WHERE clause for the key flexfield as part of the validation, you must define a parameter named `field: whereClause` in the parameter list associated with the flexfield component. See: To define a parameter list to use a WHERE clause with a key flexfield component, page 4-14.
    >
    > To use a data set identifier for the key flexfield as part of the validation, you must define a parameter named `field: datasetCol` in the parameter list associated with the flexfield component. See: To define a parameter list to use a data set identifier with a key flexfield component, page 4-15.

20. For key flexfield segment validation or key flexfield ID segment validation, specify the following validation details.

    - Enter the segment number.

21. For table validation, specify the following validation details:

    - Enter the column in the validation entity that contains the internal ID for each value.

    - Enter the column in the validation entity the contains the meaning of the value.

- Enter the column in the validation entity that contains the full description of the value.

- Enter the validation entity for the attribute, such as the view that provides access to the values.

- Enter a WHERE clause that specifies the conditions defining the data to select.

  To create a dependent list of values when the value of this attribute depends on the value of another attribute, you can include a token in the WHERE clause to reference the value of that attribute. The WHERE clause can include multiple such tokens if the list of values depends on multiple other attributes. Use the following syntax for the token for a referenced attribute:

  `$INTERFACE$.<INTERFACE_ATTRIBUTE>`

  Replace `<INTERFACE_ATTRIBUTE>` with the name of the referenced attribute.

  For example, an attribute that contains the city of birth for a person may depend on a value that contains the region of birth. In this case, the WHERE clause could be defined as follows:

  ```
  STATE_ABBREV = $INTERFACE$.REGION_OF_BIRTH
  ```

22. Select the component name that defines the list of values to use for validation. See: Managing Components, page 3-1.

23. Select the type of list of values that the component uses, either None, Pop List for a predefined static set of values, or Standard for a dynamic set of values based on Oracle E-Business Suite data.

    > **Note:** The maximum number of values that can appear in a Pop List is 255. If your list includes more than 255 values, use a Standard list of values instead.

24. Enter a prompt to appear to the left of the attribute if it appears in the header region of a document. This prompt is also used as the display name of the attribute.

25. Optionally enter a prompt to appear above the attribute if it appears in the lines region of a document.

26. Optionally enter hint text to provide users further guidance on the attribute. The hint text appears between the header region and the lines region in the document.

27. Choose Apply.

To update an interface attribute:

1. Use the Interface Attribute Definition pop-up window to update attribute details

that are not displayed in the Interfaces page.

> **Note:** To preserve consistency, you cannot update the interface attribute name, length, or attribute type. Additionally, you cannot update the data type for attributes with a data type of Number or String.

2. For an attribute that stores a date value, select the data type.

   - Date - This data type holds a date value without a time component.

   - DateTime - This data type holds a date value with a time component that is both stored in the Oracle E-Business Suite database and displayed in the desktop document according to the server time zone.

   - DateTimeTZ - This data type holds a date value with a time component that is stored in the Oracle E-Business Suite database according to the server time zone, but is displayed in the desktop document according to the user's preferred time zone specified in the Client Timezone (CLIENT_TIMEZONE_ID) profile option. Oracle Web Applications Desktop Integrator converts values with this data type to the client time zone when they are downloaded to a document and converts these values to the server time zone when they are uploaded to Oracle E-Business Suite.

     > **Additional Information:** See: User-Preferred Time Zones, *Oracle E-Business Suite Setup Guide*.

3. Select whether the attribute is required to be included in all layouts for this integrator.

4. Select whether the attribute can be summed to form a total value.

5. Select whether the attribute is read-only. A read-only attribute is placed in the context section of the desktop document and users cannot change its value.

6. Select whether the attribute is enabled for use in a mapping, to be mapped to a content column.

7. For an attribute derived from an interface table column, the Not Null checkbox indicates whether that interface table column must not be null.

8. Select the validation type for the attribute, either Descriptive Flexfield, Descriptive Flexfield Context, Descriptive Flexfield Segment, Group, Java, Key Flexfield, Key Flexfield ID, Key Flexfield ID Segment, Key Flexfield Segment, or Table. If you only want to validate the data type, length, and not null requirement for the attribute,

then you can leave the validation type blank.

9. If the attribute should be validated as part of an attribute group, select the group name. See: To manage interface attribute groups, page 2-24.

10. For descriptive flexfield validation, specify the following validation details.

    - Enter the validation entity for the attribute. The default validation entity is `oracle.apps.bne.integrator.validators.BneDFFValidator`.

    - Select the descriptive flexfield to use for validation.

    - Select Yes or No to indicate whether the descriptive flexfield segment values should be concatenated.

11. For descriptive flexfield context validation, specify the following validation details.

    - Select the descriptive flexfield to use for validation.

    - Select Yes or No to indicate whether the descriptive flexfield segment values should be concatenated.

12. For descriptive flexfield segment validation, specify the following validation details.

    - Enter the segment number.

13. For group validation or Java validation, specify the following validation details.

    - Enter the validation entity for the attribute.

14. For key flexfield or key flexfield ID validation, specify the following validation details.

    - Enter the validation entity for the attribute. The default validation entity is `oracle.apps.bne.integrator.validators.BneKFFValidator`.

    - Select the key flexfield to use for validation.

    - Select Yes or No to indicate whether the key flexfield segment values should be concatenated.

        Note: To use a WHERE clause for the key flexfield as part of the validation, you must define a parameter named `field: whereClause` in the parameter list associated with the flexfield component. See: To define a parameter list to use a WHERE clause with a key flexfield component, page 4-14.

To use a data set identifier for the key flexfield as part of the validation, you must define a parameter named `field: datasetCol` in the parameter list associated with the flexfield component. See: To define a parameter list to use a data set identifier with a key flexfield component, page 4-15.

15. For key flexfield segment validation or key flexfield ID segment validation, specify the following validation details.

    - Enter the segment number.

16. For table validation, specify the following validation details:

    - Enter the column in the validation entity that contains the internal ID for each value.

    - Enter the column in the validation entity the contains the meaning of the value.

    - Enter the column in the validation entity that contains the full description of the value.

    - Enter the validation entity for the attribute, such as the view that provides access to the values.

    - Enter a WHERE clause that specifies the conditions defining the data to select.

      To create a dependent list of values when the value of this attribute depends on the value of another attribute, you can include a token in the WHERE clause to reference the value of that attribute. The WHERE clause can include multiple such tokens if the list of values depends on multiple other attributes. Use the following syntax for the token for a referenced attribute:

      `$INTERFACE$.<`*`INTERFACE_ATTRIBUTE`*`>`

      Replace `<`*`INTERFACE_ATTRIBUTE`*`>` with the name of the referenced attribute.

      For example, an attribute that contains the city of birth for a person may depend on a value that contains the region of birth. In this case, the WHERE clause could be defined as follows:

      ```
      STATE_ABBREV = $INTERFACE$.REGION_OF_BIRTH
      ```

17. If the attribute should use a list of values defined in a component, select the component name. See: Managing Components, page 3-1.

18. Select the type of list of values that the component uses, either None, Pop List for a predefined static set of values, or Standard for a dynamic set of values based on Oracle E-Business Suite data.

> **Note:** The maximum number of values that can appear in a Pop List is 255. If your list includes more than 255 values, use a Standard list of values instead.

19. Optionally enter a prompt to appear above the attribute if it appears in the lines region of a document.

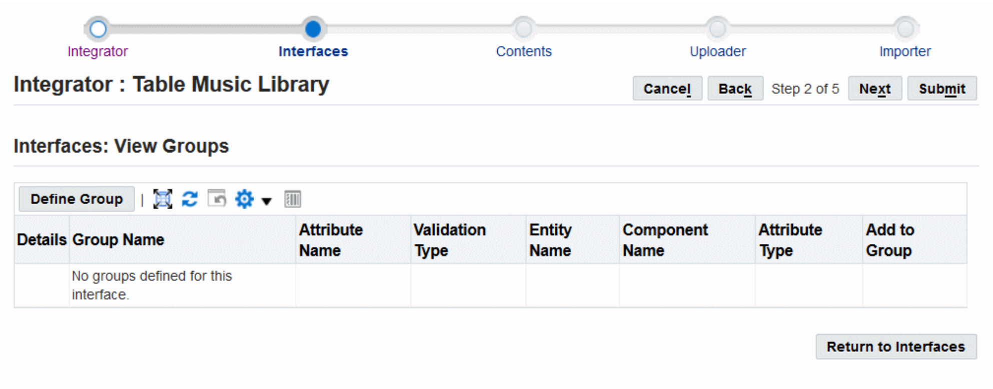
20. Optionally enter hint text to provide users further guidance on the attribute. The hint text appears between the header region and the lines region in the document.

21. Choose Apply.

To manage interface attribute groups:

1. The Interfaces: View Groups region displays a list of the interface attribute groups defined for this interface.

Interfaces: View Groups Region



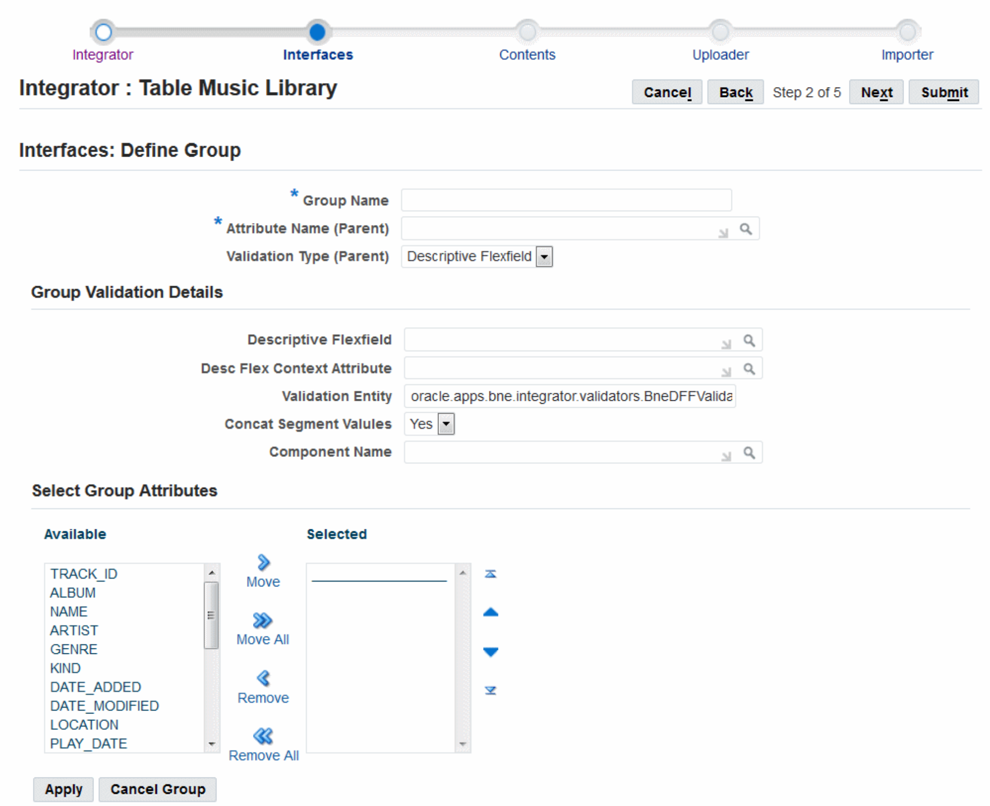
Integrator : Table Music Library

Interfaces: View Groups

Details	Group Name	Attribute Name	Validation Type	Entity Name	Component Name	Attribute Type	Add to Group
	No groups defined for this interface.						

To define a new group, choose Define Group.

Interfaces: Define Group Region



Integrator : Table Music Library

Interfaces: Define Group

* Group Name

* Attribute Name (Parent)

Validation Type (Parent): Descriptive Flexfield

Group Validation Details

Descriptive Flexfield

Desc Flex Context Attribute

Validation Entity: oracle.apps.bne.integrator.validators.BneDFFValida

Concat Segment Values: Yes

Component Name

Select Group Attributes

Available: TRACK_ID, ALBUM, NAME, ARTIST, GENRE, KIND, DATE_ADDED, DATE_MODIFIED, LOCATION, PLAY_DATE

Selected

2. In the Interfaces: Define Group region, enter a unique name for the group.

3. Select the name of the attribute to use as the parent attribute for the group. The parent attribute should be a custom attribute that is not uploaded. Instead, it serves as a placeholder for validation for the group.

4. Select the validation type associated with the parent attribute, either Descriptive Flexfield, Group, Java, Key Flexfield, or Key Flexfield ID.

5. For descriptive flexfield validation, specify the following validation details.

   - Select the descriptive flexfield to use for validation.

   - Select the interface attribute that corresponds to the context for the descriptive flexfield. This attribute should have the validation type Descriptive Flexfield Context.

     Note: For the Oracle Web Applications Desktop Integrator descriptive flexfield list of values to populate the descriptive flexfield context and segment values in the desktop document, the descriptive flexfield's reference field must either be empty or be one of the following:

- `$PROFILES$.`*`<PROFILE_OPTION_NAME>`*
- `CONTEXT`
- `CONTEXT3`
- `ACCOUNT_NUM` (used in Oracle General Ledger only)

For a descriptive flexfield whose reference field is specified as `Block.Field`, users can manually enter values in the document and upload those values. However, Oracle Web Applications Desktop Integrator does not support displaying a list of values for such a descriptive flexfield.

- Enter the validation entity for the group. The default validation entity is `oracle.apps.bne.integrator.validators.BneDFFValidator`.
- Select Yes or No to indicate whether the descriptive flexfield segment values should be concatenated.
- Select the component name that defines the list of values to use for validation. See: Managing Components, page 3-1.

6. For group validation or Java validation, specify the following validation details.
   - Enter the validation entity for the group.
   - Select the component name that defines the list of values to use for validation. See: Managing Components, page 3-1.
7. For key flexfield or key flexfield ID validation, specify the following validation details.
   - Select the key flexfield to use for validation.
   - Enter the validation entity for the group. The default validation entity is `oracle.apps.bne.integrator.validators.BneKFFValidator`.
   - Select Yes or No to indicate whether the key flexfield segment values should be concatenated.
   - Select the component name that defines the list of values to use for validation. See: Managing Components, page 3-1.

8. Select the attributes you want in the Available list and move them to the Selected list.

9. Choose Apply.

10. In the Interfaces: View Groups region, to display the attributes that belong to a group, choose the Details icon for the group.

11. To remove an attribute from the group, choose the Ungroup icon for the attribute in the group attributes list.

12. To add more attributes to the group, choose the Add to Group icon for the group in the list of groups.

13. After you finish managing interface attribute groups, choose Return to Interfaces.

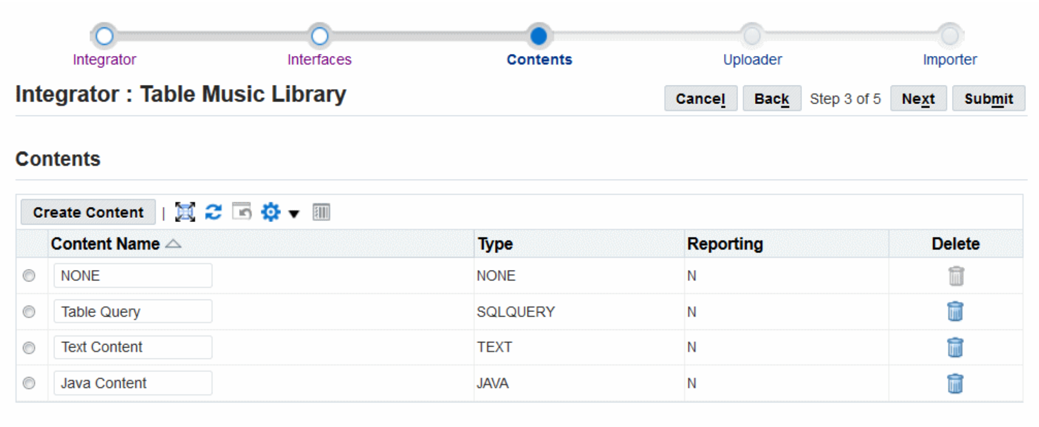
To define contents:

A *content* is a set of metadata that determines what data the integrator uses to populate the desktop document that it creates. Use the Contents page of the integrator definition wizard to manage and create contents.

- For a reporting-only integrator, you must define at least one SQL query or Java content to specify the Oracle E-Business Suite data that the integrator downloads into the desktop document.

- For an integrator that can upload data to Oracle E-Business Suite, you can optionally define SQL query or Java contents that download data from Oracle E-Business Suite into the document, or text file contents that populate the document with data from the specified text file. You can also choose not to define any contents to create a blank document. In this case users must enter all data into the document manually.

Contents Page



Integrator : Table Music Library

Contents

Content Name	Type	Reporting	Delete
NONE	NONE	N	
Table Query	SQLQUERY	N	
Text Content	TEXT	N	
Java Content	JAVA	N	

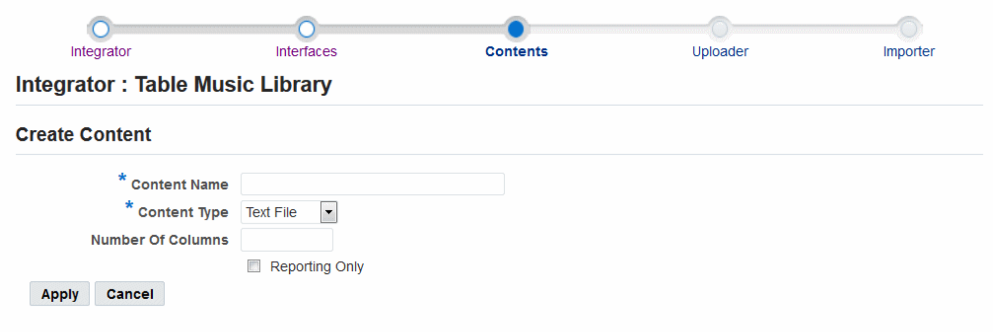
1. The Contents page displays the list of contents defined for the integrator.

   - To update the name of a content, enter the new name in the Content Name field.

   - To delete a content, choose the delete icon for that content.

   - To create a new content, select the Create Content button. The Create Content region appears.

Create Content Region



Integrator : Table Music Library

Create Content

* Content Name

* Content Type: Text File

Number Of Columns

□ Reporting Only

2. In the Create Content for Integrator region, enter the content name.

3. Select the content type, either Java, SQL query, or text file.

   - For a Java content, enter the Java class that returns the Oracle E-Business Suite data you want to download, and the number of arguments for that Java class. Oracle E-Business Suite Desktop Integration Framework creates a content column for each argument. You can include logic in your Java class to perform additional processing on the downloaded data before inserting the data into the desktop document.

- For a SQL query content, enter in the Query field the query code that selects the Oracle E-Business Suite data you want to download.

  > **Note:** The maximum length of the query is 4,000 characters. If you enter a longer query, Oracle E-Business Suite Desktop Integration Framework truncates the query after 4,000 characters. It is recommended that you remove any extra spaces from the query if necessary to reduce its length.

  Choose the Test Query button to validate the query syntax. If the query is not valid, Oracle E-Business Suite Desktop Integration Framework displays the Oracle syntax error that was found in order to help you correct the query.

  When the SQL query content is saved, Oracle E-Business Suite Desktop Integration Framework creates a content column for each column selected in the query.

  > **Note:** The SQL query for a content can include additional parameters whose values can be modified or entered by the end user when the integrator is run. The query can also include conditions based on environment variables or on functions such as SYSDATE. When preparing the query to be run, Oracle Web Applications Desktop Integrator parses the query and replaces any parameters and environment variables with the appropriate values.
  >
  > The syntax for a parameter is:
  >
  > `$param$.<parameter_name>`
  >
  > The syntax for an environment variable is:
  >
  > `$env$.<variable>`
  >
  > The environment variables supported for use within a SQL query are:
  >
  > - `$env$.appid`
  > - `$env$.userid`
  > - `$env$.language`
  > - `$env$.respid`
  > - `$env$.sessionid`
  >
  > This sample query shows an example of including an

environment variable and a parameter in the query conditions.

```
SELECT
   APPLICATION_ID,
   INTEGRATOR_CODE,
   ENABLED_FLAG,
   UPLOAD_PARAM_LIST_APP_ID,
   UPLOAD_PARAM_LIST_CODE,
   UPLOAD_SERV_PARAM_LIST_APP_ID,
   UPLOAD_SERV_PARAM_LIST_CODE,
   IMPORT_PARAM_LIST_APP_ID,
   IMPORT_PARAM_LIST_CODE,
   UPLOADER_CLASS,
   DATE_FORMAT,
   IMPORT_TYPE,
   CREATE_DOC_LIST_APP_ID,
   CREATE_DOC_LIST_CODE,
   NEW_SESSION_FLAG,
   USER_NAME,
   UPLOAD_TITLE_BAR,
   UPLOAD_HEADER,
   LAYOUT_RESOLVER_CLASS,
   LAYOUT_VERIFIER_CLASS,
   SESSION_CONFIG_CLASS,
   SESSION_PARAM_LIST_APP_ID,
   SESSION_PARAM_LIST_CODE,
   DISPLAY_FLAG,
   SOURCE
FROM BNE_INTEGRATORS_VL WHERE APPLICATION_ID =
$env$.appid AND USER_NAME LIKE $param$.
INTG_USER_NAME_PARAM
```

- For a text file content, enter the number of columns that the text file contains. Oracle E-Business Suite Desktop Integration Framework creates a content column for each column in the text file.

4. If you want to use this content only for downloading data from Oracle E-Business Suite to the desktop, select the Reporting Only option.

5. Choose Apply.

6. In the Contents page, select a content in the list to display its details including the content columns and parameter list.

7. Expand the Content Columns region to view the columns that were automatically created for the content. You can optionally update the display name for each column. You can also select the Read Only option to specify that the column is read-only and cannot be updated.

   You can also optionally add columns to the content. For example, you can add columns if the Java class has been updated to include additional arguments, if you need to update the SQL query to select additional columns, or if the text file has been updated to include additional columns.

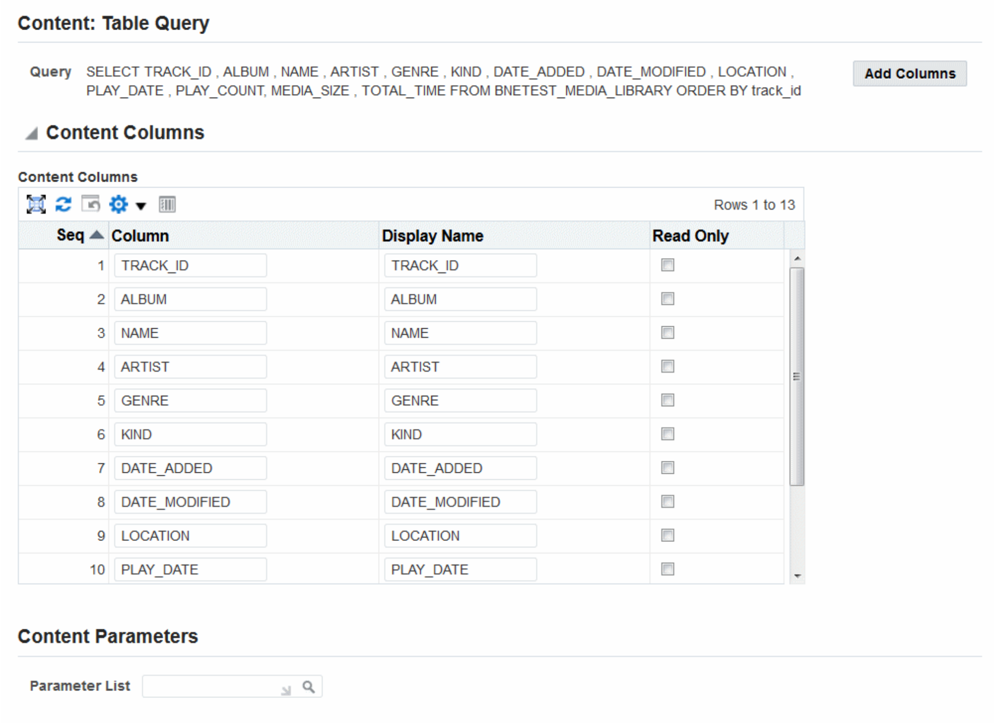
- For a Java content or a text file content, choose Add Column. The new column will be added at the end of the list of columns for the content.

- For a SQL query content, select Add Columns. In the popup window that appears, review the existing SQL query and enter a comma-separated list of column names or expressions that you want to add to the SELECT clause of the query. Then choose Save. The new columns will be added at the end of the SELECT clause.

Oracle E-Business Suite Desktop Integration Framework displays the details for each newly added column at the end of the list in the Content Columns region.

After you add a column to a content for an existing integrator, you should also update the layout for the integrator as needed to accommodate the new column. If this integrator allows uploading data to Oracle E-Business Suite, ensure that you also update the interface and mapping as needed. For a reporting-only integrator, Oracle E-Business Suite Desktop Integration Framework automatically updates the default placeholder interface and adds the new mapping line when you add a column to a content.

Contents Page - Content Columns Region and Content Parameters Region



Content: Table Query

Query SELECT TRACK_ID , ALBUM , NAME , ARTIST , GENRE , KIND , DATE_ADDED , DATE_MODIFIED , LOCATION , PLAY_DATE , PLAY_COUNT, MEDIA_SIZE , TOTAL_TIME FROM BNETEST_MEDIA_LIBRARY ORDER BY track_id

Content Columns

Content Columns

Seq	Column	Display Name	Read Only
1	TRACK_ID	TRACK_ID	☐
2	ALBUM	ALBUM	☐
3	NAME	NAME	☐
4	ARTIST	ARTIST	☐
5	GENRE	GENRE	☐
6	KIND	KIND	☐
7	DATE_ADDED	DATE_ADDED	☐
8	DATE_MODIFIED	DATE_MODIFIED	☐
9	LOCATION	LOCATION	☐
10	PLAY_DATE	PLAY_DATE	☐

Content Parameters

Parameter List

8. If the content requires additional parameters, select a parameter list in the Content Parameters region. See: Managing Parameters, page 4-1.

   Note: For a text file content, you can optionally define parameters that let you specify default values for the character to use as the file delimiter, whether to ignore consecutive delimiters, and the line at which to start importing. If you choose to use these parameters, you must include them in the parameter list associated with the text file content. See: To define a parameter list to specify handling for a text file content, page 4-16.

   Users can override the default values for these parameters when creating a document. See: Creating a Spreadsheet, *Oracle Web Applications Desktop Integrator Implementation and Administration Guide*.

9. To proceed to the Uploader page, select Next, or to save this integrator definition without defining an uploader or importer, select Submit.

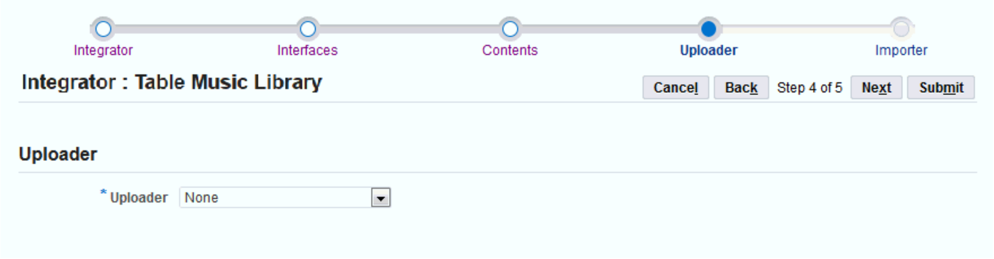
To define an uploader:

An *uploader* is a set of metadata that determines what options are available to a user in the upload parameters window when the user uploads data from a desktop document to Oracle E-Business Suite. Use the Uploader page of the integrator definition wizard to create and manage the uploader for an integrator.

> **Note:** If you selected the Reporting Only option for this integrator, then you do not need to define an uploader.

1. If an uploader has not yet been defined for the integrator, then the Uploader page displays the Uploader field.

   Uploader Page with Uploader Field

   Integrator : Table Music Library

   Uploader

   * Uploader: None

   In the Uploader field, select the method by which to create the uploader.

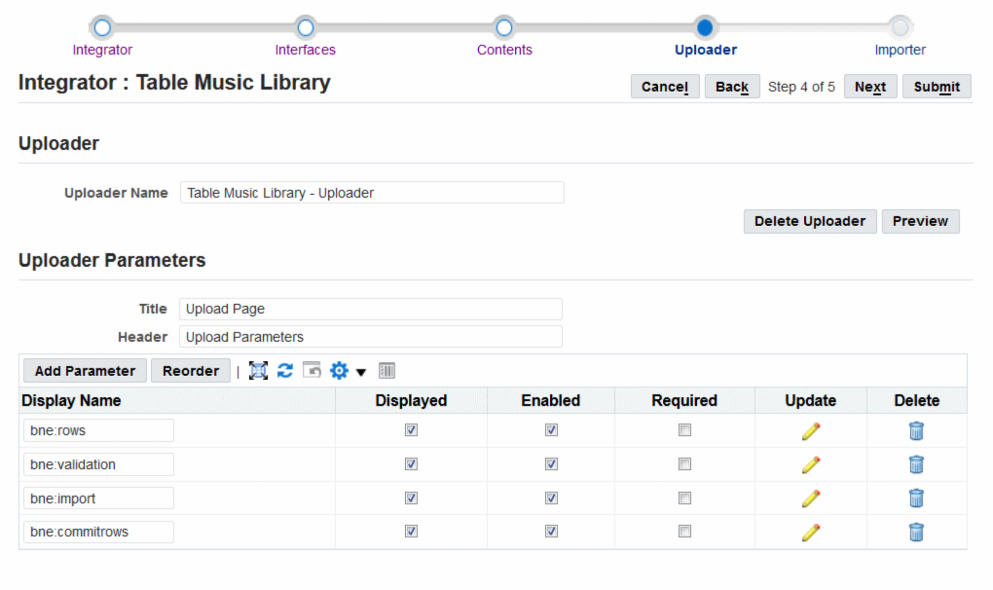
   - None - Select this option if you do not want to use this integrator to upload data from a desktop document to Oracle E-Business Suite. In this case no uploader is created. To proceed to the Importer page, select Next, or to save this integrator definition without defining an importer, select Submit.

   - From Template - Use the standard template provided by Oracle E-Business Suite Desktop Integration Framework as the basis for the uploader definition. After selecting this option, choose the Create button to create the uploader.

   - Copy From Existing Definition - Use the definition of an existing uploader from another integrator as the basis for the uploader definition. After selecting this option, select the existing uploader definition that you want to copy. Then choose the Create button to create the uploader.

     > **Note:** You can copy an uploader definition only from another integrator that belongs to the same application as the integrator you are defining.

Uploader Page



Integrator : Table Music Library

Uploader

Uploader Name: Table Music Library - Uploader

Uploader Parameters

Title: Upload Page

Header: Upload Parameters

Display Name	Displayed	Enabled	Required	Update	Delete
bne:rows	☑	☑	□	✎	🗑
bne:validation	☑	☑	□	✎	🗑
bne:import	☑	☑	□	✎	🗑
bne:commitrows	☑	☑	□	✎	🗑

2. In the Uploader page, specify the display name of the uploader. The default name is `<Integrator Name> - Uploader`. You can optionally enter a different name.

3. To delete the existing uploader definition for the integrator, choose the Delete Uploader button. The Uploader page then displays the Uploader field to let you either create a new uploader or proceed without defining an uploader.

4. To preview how the upload parameters window will be displayed to users based on the current uploader definition, choose the Preview button.

5. In the Uploader Parameters region, specify the title that should appear in the browser title bar of the upload parameters window. The default title is `Upload Page`. You can optionally enter a different title.

6. Specify the header that should appear within the upload parameters window. The default header is `Upload Parameters`. You can optionally enter a different header.

7. If you created this uploader using the standard template, then Oracle E-Business Suite Desktop Integration Framework automatically creates the following parameters:

   - Rows to Upload (`bne:rows`) - Determines whether to upload all rows in the spreadsheet, or only those rows marked with an upload flag. Oracle Web Applications Desktop Integrator marks all changed rows with an upload flag.

- Validate Before Upload (`bne:validation`) - Determines whether to validate data before uploading the data to Oracle E-Business Suite.

- Automatically Submit Import (`bne:import`) - Determines whether to submit an import process to move the uploaded data from Oracle E-Business Suite interface tables to base application tables automatically after the upload. If you include this parameter, then you should define an importer for this integrator to specify the import process that will be submitted.

- Commit Rows (`bne:commitrows`) - Determines whether to commit the uploaded rows only if all the rows are valid, or commit each uploaded row that is valid, even if other rows are invalid.

If you created this uploader as a copy of an existing uploader definition, then Oracle E-Business Suite Desktop Integration Framework automatically creates copies of all the parameters defined for the existing uploader.

You can update or delete any of the automatically created parameters or define additional parameters as needed.

8. You can update some parameter details directly in the Uploader Parameters region.

   - Enter the display name for the parameter.

   - Select whether the parameter is displayed in the upload parameters window.

   - Select whether the parameter is enabled for users to enter a value in the upload parameters window.

   - Select whether users are required to enter a value for the parameter in the upload parameters window.

   You can also choose the update icon or the view icon for a parameter to view and update additional parameter details. See: To define an upload parameter, page 2-36 .

9. To delete a parameter, choose the delete icon for that parameter.

10. To define a new upload parameter, choose the Add Parameter button. See: To define an upload parameter, page 2-36.

    > **Note:** If you plan to define an importer for this integrator, you should define the Automatically Submit Import (`bne:import`) upload parameter within the uploader to enable the import process to be submitted automatically after the upload.

11. To change the order in which the parameters appear in the upload parameters

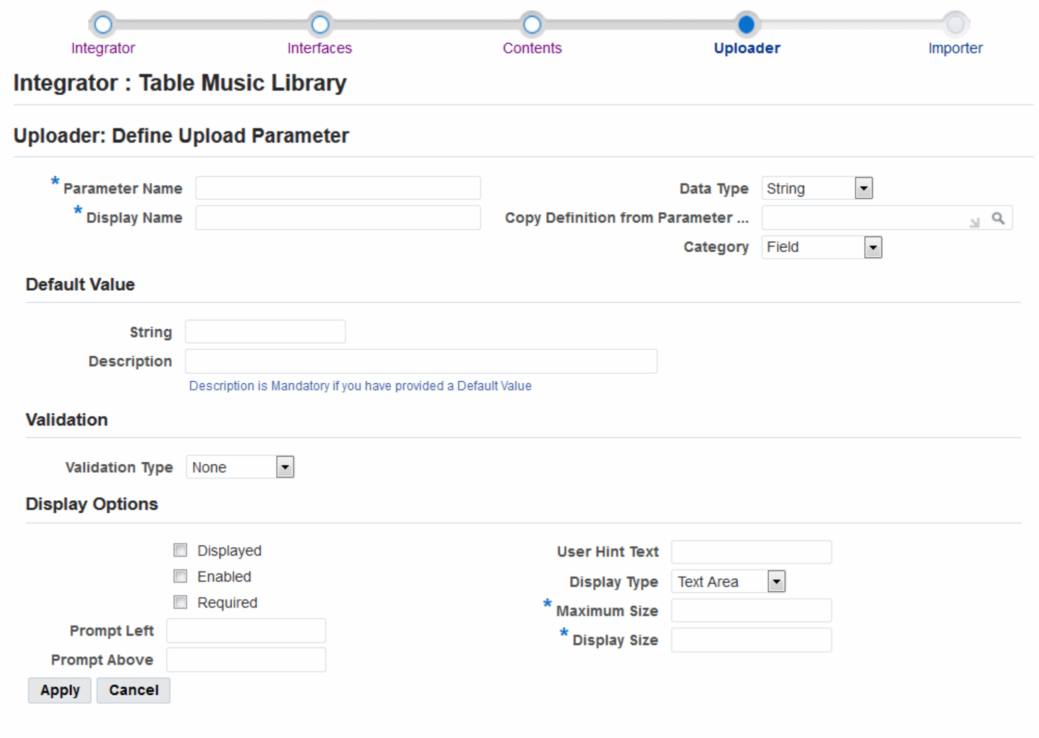
window, choose the Reorder button. Select one or more parameters in the list, and use the arrow buttons to move those parameters to the desired position. Then choose Apply.

> **Note:** You can only reorder parameters when the list contains two or more parameters.

12. To proceed to the Importer page, select Next, or to save this integrator definition without defining an importer, select Submit.

To define an upload parameter:

Uploader: Define Upload Parameter Page



Integrator : Table Music Library

Uploader: Define Upload Parameter

* Parameter Name

* Display Name

Data Type: String

Copy Definition from Parameter ...

Category: Field

Default Value

String

Description

Description is Mandatory if you have provided a Default Value

Validation

Validation Type: None

Display Options

□ Displayed

□ Enabled

□ Required

Prompt Left

Prompt Above

User Hint Text

Display Type: Text Area

* Maximum Size

* Display Size

1. In the Uploader: Define Upload Parameter page, optionally select an existing parameter definition to copy. The remaining fields in the page then display the values copied from that parameter. You can update these values as needed.

2. Enter a unique internal name for the parameter.

3. Enter the display name for the parameter.

4. Select the data type of the parameter: String, Number, Date, Boolean, or Key

Flexfield.

5. Select the category to which the parameter belongs.

   - Miscellaneous
   - Position
   - Appearance
   - Behavior
   - Data
   - Field

6. Optionally enter a default value of the appropriate data type for the parameter.

   - For Boolean parameter, select either Yes or No as the default value.
   - For a key flexfield parameter, select the flexfield structure. The Default Value region displays the ID of the application to which the flexfield belongs and the numerical ID for the flexfield.

7. If you entered a default value, enter a description for that value.

8. Select the validation type for the parameter: None, Value Set, Fnd Lookup, or Bne Query.

9. Depending on the selected validation type, enter the value set, lookup type, or SQL query against which you want to validate the parameter value.

   For a SQL query, select whether you want to use a simple or complex query.

   - For a simple query, you can enter the following common query details and let Oracle E-Business Suite Desktop Integration Framework generate the complete query syntax for you.
     - A list of one or more tables from which to select data, separated by commas
     - The ID column to select, and an alias for that column
     - The meaning column to select, and an alias for that column
     - The description column to select, and an alias for that column
     - A list of additional columns to select
     - A WHERE clause that specifies the conditions defining the data to select

    - A list of the columns by which the query results should be ordered

  - For a complex query, you must enter the complete query syntax yourself.

  After entering the query, choose the Validate Query button to validate the query syntax.

10. Select whether the parameter is displayed in the upload parameters window.

11. Select whether the parameter is enabled for users to enter a value in the upload parameters window.

12. Select whether users are required to enter a value for the parameter in the upload parameters window.

13. Optionally enter a prompt to appear to the left of the parameter.

14. Optionally enter a prompt to appear above the parameter.

15. Optionally enter a hint to display to provide users further guidance on the parameter.

16. Select the display type for the parameter.

    - For a string parameter, you can select Check Box, Radio Button, List Box, Text Area, or Password.

    - For a number or date parameter, you can select Radio Button, List Box, Text Area, or Password.

    - For a Boolean parameter, you can select Check Box, Radio Button, List Box, or Text Area.

    - A key flexfield parameter must use the Text Area display type.

17. If you selected the display type Radio Button, select whether the radio buttons should be displayed in a vertical or horizontal style.

18. Enter the maximum size for the parameter value.

19. Enter the display size for the parameter value. The display size must be equal to or less than the maximum size.

20. Choose Apply.

To define an importer:

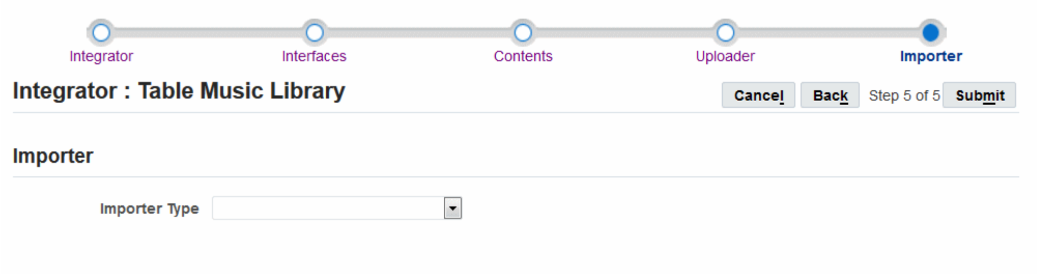
An *importer* is a set of metadata that determines how Oracle Web Applications Desktop

Integrator moves uploaded data from Oracle E-Business Suite interface tables to base application tables. Use the Importer page of the integrator definition wizard to define the importer for an integrator.

Ensure that you define the uploader for an integrator before defining an importer. Additionally, ensure that you define the Automatically Submit Import (`bne:import`) upload parameter within the uploader to enable the import process to be submitted automatically after the upload.

> **Note:** If you selected the Reporting Only option for this integrator, then you do not need to define an importer.

Importer Page: Importer Type



Integrator : Table Music Library

Importer

Importer Type

1. In the Importer page, select the importer type.

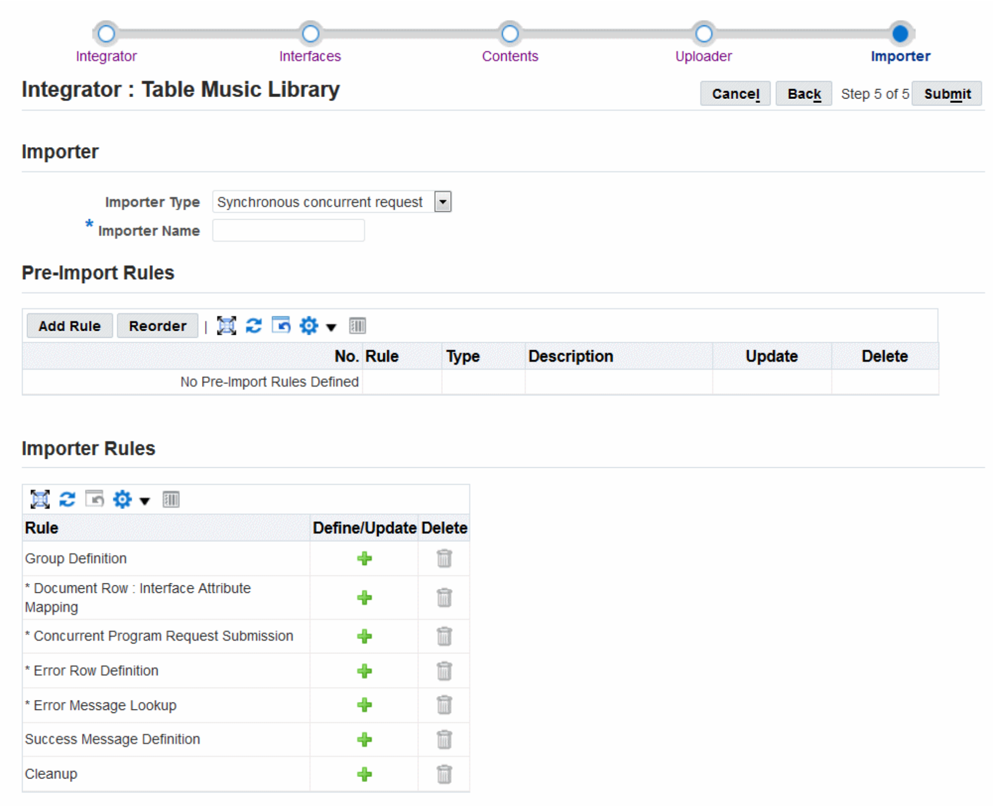
   - Asynchronous concurrent request - Oracle Web Applications Desktop Integrator submits the specified concurrent program after completing the upload and returns the concurrent request ID to the user, but does not wait for the import program to complete. This importer type reduces user wait time by deferring the import process, and gives system administrators more flexibility in load balancing the resources involved in importing data. However, users may need to verify later that the import completed successfully.

     If you select this importer type, it is recommended that you define pre-import rules to validate the data at the time of upload. If the pre-validation fails, then Oracle Web Applications Desktop Integrator does not upload the data to the interface tables or invoke the import program, but returns an error message to the user instead.

   - Synchronous concurrent request - Oracle Web Applications Desktop Integrator submits the specified concurrent program after completing the upload, waits for the program to complete, and then returns the results to the user. This importer type gives users immediate feedback about the success of the import, but can increase the load on resources as well as user wait time.

   - PL/SQL API - Oracle Web Applications Desktop Integrator calls the specified API after completing the upload, waits for the API processing to complete, and then returns the results to the user. This importer type gives users immediate feedback about the success of the import and partly reduces user wait time by bypassing the concurrent processing queue. However, it can increase the load on resources.

Importer Page



Integrator : Table Music Library

Importer

Importer Type: Synchronous concurrent request

* Importer Name:

Pre-Import Rules

No.	Rule	Type	Description	Update	Delete
No Pre-Import Rules Defined					

Importer Rules

Rule	Define/Update	Delete
Group Definition	✚	🗑
* Document Row : Interface Attribute Mapping	✚	🗑
* Concurrent Program Request Submission	✚	🗑
* Error Row Definition	✚	🗑
* Error Message Lookup	✚	🗑
Success Message Definition	✚	🗑
Cleanup	✚	🗑

2. Enter a unique name for the importer.

3. If you want to perform validation or other processing before the import process, choose the Add Rule button in the Pre-Import Rules region. See: To define a pre-import rule, page 2-42.

4. To update a pre-import rule, choose the update icon for that rule. See: To define a pre-import rule, page 2-42.

5. To delete a rule, choose the delete icon for that rule.

6. The Importer Rules region displays a list of possible rule types for the import process itself, according to the importer type. An asterisk marks required rule types.

   - To define an importer rule, choose the define/update icon for that rule type.

  - Group Definition - Identifies records in the interface table that should be imported together as a group. All other importer rules, except group definition rules and document row - interface attribute mapping rules, are repeated for each group of records in the interface table. See: To define a group definition importer rule, page 2-44.

  - Document Row: Interface Attribute Mapping - Maps rows in the desktop spreadsheet to records in the interface table. This rule type is required for a synchronous concurrent request importer and optional for other importer types. See: To define a document row - interface attribute mapping importer rule, page 2-45.

  - Concurrent Program Request Submission - Specifies the concurrent program to perform the import. This rule type is available and required only for an asynchronous or synchronous concurrent request importer. See: To define a concurrent program request submission importer rule, page 2-46.

  - PL/SQL API call - Specifies the PL/SQL API to perform the import. This rule type is available and required only for a PL/SQL API importer. See: To define a PL/SQL API call importer rule, page 2-47.

  - Error Row Definition - Identifies records that were not imported successfully. This rule type is available and required only for a synchronous concurrent request or a PL/SQL API importer. See: To define an error row definition importer rule, page 2-48.

  - Error Message Lookup - Retrieves the error message to return to the user for records that were not imported successfully. This rule type is available and required only for a synchronous concurrent request or a PL/SQL API importer. See: To define an error message lookup importer rule, page 2-50.

  - Success Message Definition - Specifies a message to display to the user when the import process is submitted successfully. See: To define a success message definition importer rule, page 2-51.

  - Cleanup - Specifies cleanup processing to perform if errors occurred during any of the previous importer rules. See: To define a cleanup importer rule, page 2-51.

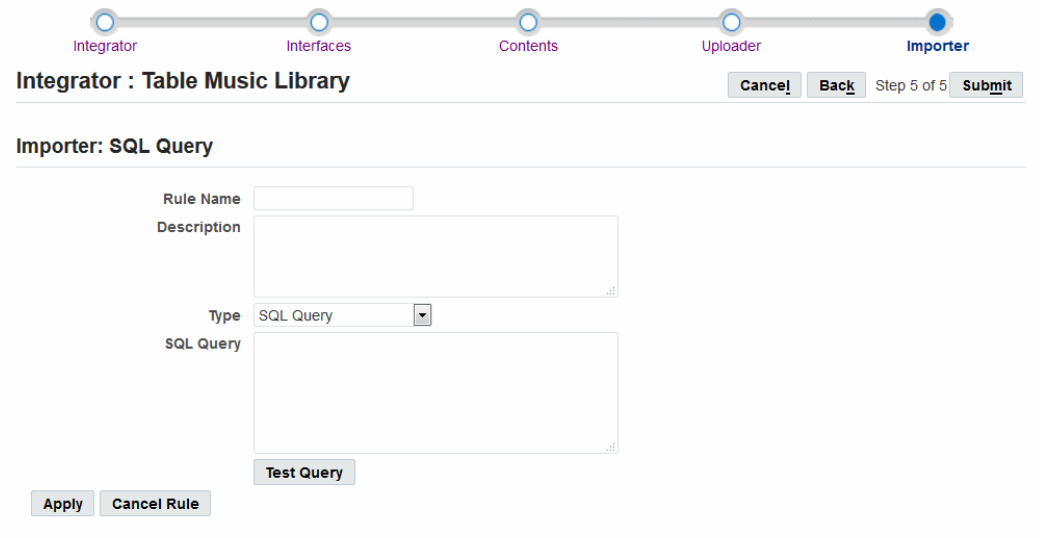
- To delete a rule definition, choose the delete icon for that rule type. The rule type still appears in the list to let you create a new rule definition later if you choose.

7. To save the integrator definition, select Submit.

To define a pre-import rule:

Use pre-import rules to specify validation or other processing that you want to perform on the data being uploaded before importing the data into application base tables. A pre-import rule can perform a SQL query, call a PL/SQL function or procedure, or retrieve the next value of a database sequence.

Importer Page: Pre-Import Rule



Integrator : Table Music Library

Importer: SQL Query

Rule Name

Description

Type SQL Query

SQL Query

1. Enter a unique name for the pre-import rule.

2. Enter a description for the rule.

3. Select the rule type: SQL Query, PL/SQL API Function, PL/SQL API Procedure, or Sequence.

4. For a SQL query rule, enter the query. Choose the Test Query button to validate the query syntax.

5. For a PL/SQL function rule, enter the package name and function name, and specify whether the API returns an error message or an FND message code if it encounters an error.

6. For a PL/SQL procedure rule, enter the package name and procedure name, and specify whether the API returns an error message or an FND message code if it encounters an error.

7. For a sequence rule, enter the sequence name. You can optionally also enter a unique reference name by which subsequent rules can refer to this sequence.

8. Choose Apply. For a SQL query, PL/SQL function, or PL/SQL procedure rule, Oracle E-Business Suite Desktop Integration Framework displays additional details depending on the rule type.

9. For a SQL query rule, the page displays a list of the columns selected by the query. You can optionally enter a unique reference name for each column by which subsequent rules can refer to the column.

   The page also displays a list of the parameters used in the query.

   - You can optionally enter a unique reference name by which subsequent rules can refer to the parameter.

   - Select the data type for the parameter.

   - Specify the source from which the parameter value will be obtained: Environment Variables, Import, or Upload Parameters.

   - Specify the value based on the selected source. If the source is environment variables, enter the variable to use for the value, such as User ID, Application, Responsibility ID, or Language. If the source is Import, enter the interface attribute or reference parameter to use for the value. If the source is upload parameters, enter the upload parameter to use for the value.

10. For a PL/SQL function rule, if the function returns more details about any error that it encounters, select the Returns Error Information option.

11. For a PL/SQL function or procedure rule, specify whether the API returns an error message or an FND message code if it encounters an error.

12. For a PL/SQL function or procedure rule, the page displays a list of the parameters for the function or procedure.

    - You can optionally enter a unique reference name by which subsequent rules can refer to the parameter.

    - If the parameter is used to retrieve an error message, select the Error Message checkbox.

    - Optionally enter a default value for the parameter.

    - Specify the source from which the parameter value will be obtained: Environment Variables, Import, Interface Table, or Upload Parameters.

    - Specify the value based on the selected source. If the source is environment variables, enter the variable to use for the value, such as User ID, Application, Responsibility ID, or Language. If the source is Import, enter the interface attribute or reference parameter to use for the value. If the source is Interface

Table, enter the interface table details. If the source is upload parameters, enter the upload parameter to use for the value.

- Optionally specify a parameter list to use for key-value lookups.
- Optionally enter a SQL clause to modify the value, such as `TO_CHAR()`.

13. Choose Apply.

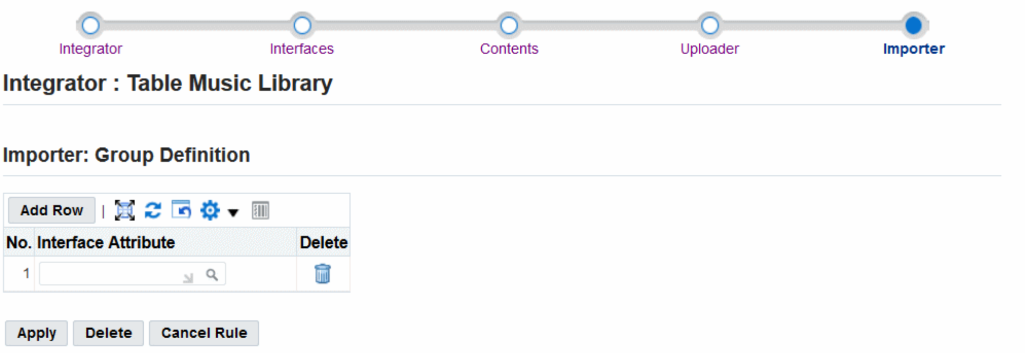
    Note: If you no longer require this rule definition, you can choose the Delete button to remove it.

To define a group definition importer rule:

Use a group definition rule to identify records in the interface table that should be imported together as a group. All other importer rules, except group definition rules and document row - interface attribute mapping rules, are repeated for each group of records in the interface table. An asynchronous or synchronous concurrent request importer submits a separate concurrent request for each record group

To identify the records that form a group, specify one or more interface attributes that together serve as a unique key for the group. All records in the interface table that have the same values in those interface attributes are treated as part of the group.

Importer Page: Group Definition Rule



Integrator : Table Music Library

Importer: Group Definition

No.	Interface Attribute	Delete
1		

1. Choose the Add Row button.
2. Select an interface attribute to identify the records in the group.
3. Optionally repeat steps 1 and 2 to add further interface attributes to identify the records in the group. Each interface attribute you add narrows the group definition.

   To remove an interface attribute from the group definition, choose the delete icon for that attribute.

4. Choose Apply.

> **Note:** If you no longer require this rule definition, you can choose the Delete button to remove it.

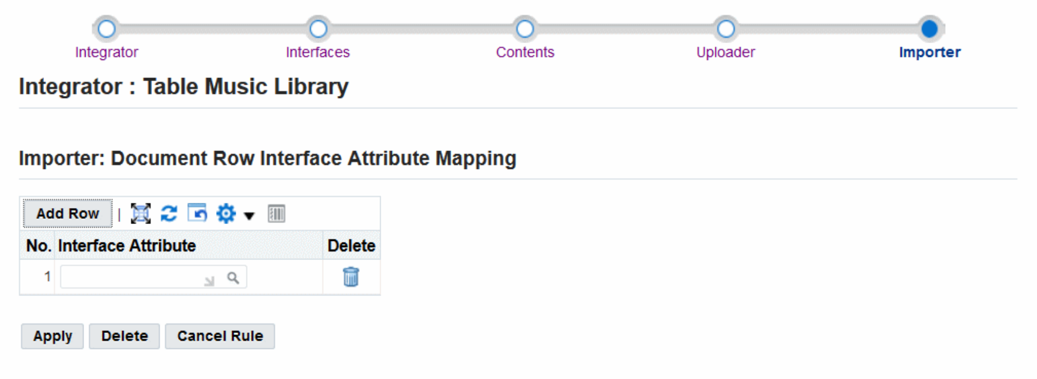
To define a document row: interface attribute mapping importer rule

Use a document row: interface attribute mapping rule to map rows in the desktop spreadsheet to records in the interface table. To define the mapping, specify one or more interface attributes that together serve as a unique key to match each spreadsheet row with the corresponding interface table record. In most cases the interface attributes that you specify should be the columns that make up the primary key of the interface table.

You can use this mapping to return individual error messages for any records that fail during import.

> **Note:** If there are no columns in the interface table that can identify a one-to-one mapping between the spreadsheet rows and the interface table records, then you cannot report individual error messages for any failed records, only a general error message for the import process as a whole.

Importer Page: Document Row Interface Attribute Mapping Rule



Integrator : Table Music Library

Importer: Document Row Interface Attribute Mapping

No.	Interface Attribute	Delete
1		

1. Choose the Add Row button.

2. Select an interface attribute by which to identify a record.

3. Repeat steps 1 and 2 to add further interface attributes as needed to identify a record uniquely.

To remove an interface attribute from the unique key, choose the delete icon for that attribute.

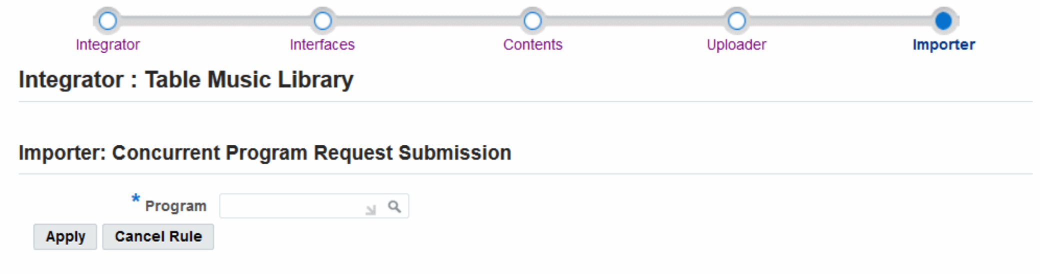
4. Choose Apply.

> **Note:** If you no longer require this rule definition, you can choose the Delete button to remove it.

To define a concurrent program request submission importer rule:

Use a concurrent program request submission rule to specify what concurrent program to submit to perform the import, as well as how to obtain the parameter values for the program.

Importer Page: Concurrent Program Request Submission Rule


Integrator
Interfaces
Contents
Uploader
Importer
Integrator : Table Music Library
Importer: Concurrent Program Request Submission
* Program
Apply
Cancel Rule

1. Select the concurrent program to perform the import.

2. Choose Apply.

3. The page displays a list of the parameters for the program.

   - You can optionally enter a unique reference name by which subsequent rules can refer to the parameter.

   - If the parameter is used to retrieve an error message, select the Error Message checkbox.

   - Optionally enter a default value for the parameter.

   - Specify the source from which the parameter value will be obtained: Environment Variables, Import, or Interface Table, or Upload Parameters.

   - Specify the value based on the selected source. If the source is environment variables, enter the variable to use for the value, such as User ID, Application, Responsibility ID, or Language. If the source is Import, enter the interface attribute or reference parameter to use for the value. If the source is Interface

Table, enter the interface table details. If the source is upload parameters, enter the upload parameter to use for the value.

- Optionally specify a parameter list to use for key-value lookups.
- Optionally enter a SQL clause to modify the value, such as `TO_CHAR()`.

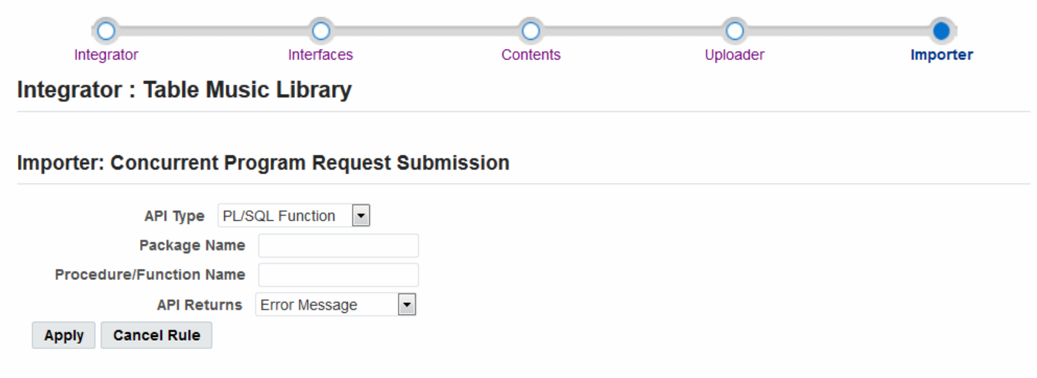
4. Choose Apply.

   Note: If you no longer require this rule definition, you can choose the Delete button to remove it.

To define a PL/SQL API call importer rule:

Use a PL/SQL API call rule to specify what PL/SQL API to call to perform the import, as well as how to obtain the parameter values for the API.

Importer Page: PL/SQL API Call Rule



Integrator : Table Music Library

Importer: Concurrent Program Request Submission

API Type	PL/SQL Function
Package Name	
Procedure/Function Name	
API Returns	Error Message

1. Select the API type, either PL/SQL Function or PL/SQL Procedure.
2. For a PL/SQL function, enter the package name and function name, and specify whether the API returns an error message or an FND message code if it encounters an error.
3. For a PL/SQL procedure, enter the package name and procedure name, and specify whether the API returns an error message or an FND message code if it encounters an error.
4. Choose Apply. Oracle E-Business Suite Desktop Integration Framework displays additional details for the API.
5. For a PL/SQL function, if the function returns more details about any error that it encounters, select the Returns Error Information option.

6. Specify whether the API returns an error message or an FND message code if it encounters an error.

7. The page displays a list of the parameters for the function or procedure.

   - You can optionally enter a unique reference name by which subsequent rules can refer to the parameter.

   - If the parameter is used to retrieve an error message, select the Error Message checkbox.

   - Optionally enter a default value for the parameter.

   - Specify the source from which the parameter value will be obtained: Environment Variables, Import, Interface Table, or Upload Parameters.

   - Specify the value based on the selected source. If the source is environment variables, enter the variable to use for the value, such as User ID, Application, Responsibility ID, or Language. If the source is Import, enter the interface attribute or reference parameter to use for the value. If the source is Interface Table, enter the interface table details. If the source is upload parameters, enter the upload parameter to use for the value.

   - Optionally specify a parameter list to use for key-value lookups.

   - Optionally enter a SQL clause to modify the value, such as `TO_CHAR()`.

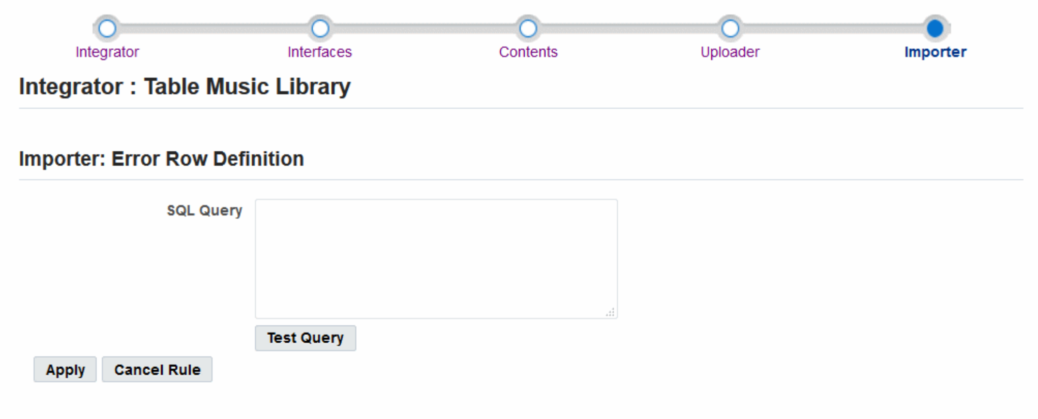
8. Choose Apply.

   > **Note:** If you no longer require this rule definition, you can choose the Delete button to remove it.

To define an error row definition importer rule:

Use an error row definition rule to identify records that were not imported successfully. To identify the failed records, specify a SQL query that selects from the interface table. It is recommended to select all columns in the interface table by beginning the query with `SELECT *` to ensure that all necessary values are retrieved.

Importer Page: Error Row Definition Rule



Integrator : Table Music Library

Importer: Error Row Definition

SQL Query

1. Enter the SQL query to identify failed records in the interface table. Choose the Test Query button to validate the query syntax.

2. Choose Apply.

3. Oracle E-Business Suite Desktop Integration Framework displays a list of the columns selected by the query. You can optionally enter a unique reference name for each column by which subsequent rules can refer to the column.

   The page also displays a list of the parameters used in the query.

   - You can optionally enter a unique reference name for the parameter by which subsequent rules can refer to the parameter.

   - Select the data type for the parameter.

   - Specify the source from which the parameter value will be obtained: Environment Variables, Import, Interface Table, or Upload Parameters.

   - Specify the value based on the selected source. If the source is environment variables, enter the variable to use for the value, such as User ID, Application, Responsibility ID, or Language. If the source is Import, enter the interface attribute or reference parameter to use for the value. If the source is Interface Table, enter the interface table details. If the source is upload parameters, enter the upload parameter to use for the value.

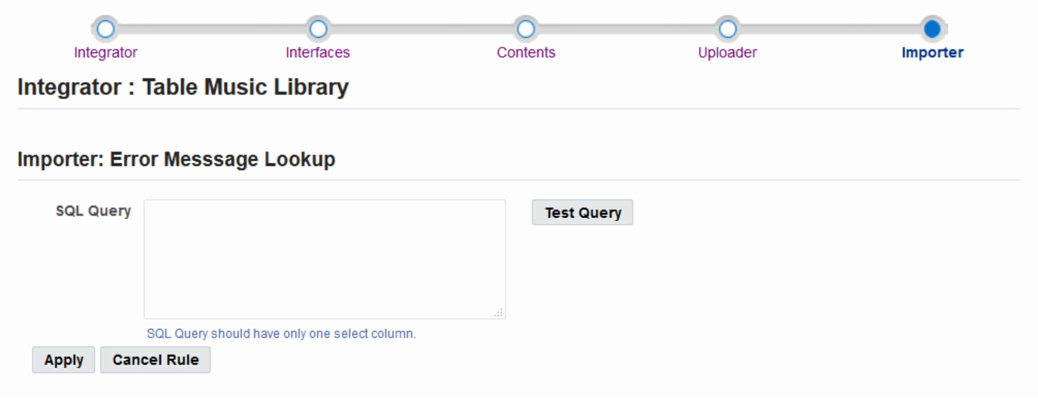
4. Choose Apply.

> **Note:** If you no longer require this rule definition, you can choose the Delete button to remove it.

To define an error message lookup importer rule:

Use an error message lookup rule to retrieve the error message to return to the user for records that were not imported successfully. Specify a SQL query that selects only one column, which should contain the error message. Oracle Web Applications Desktop Integrator runs the SQL query for each failed record in the interface table. You can reference the unique key defined in the document row - interface attribute mapping rule to match the individual failed records, with the spreadsheet rows, and use any error codes in the status column in the interface table for each record to retrieve the appropriate error message.

Importer Page: Error Message Lookup Rule



Integrator : Table Music Library

Importer: Error Messsage Lookup

SQL Query

SQL Query should have only one select column.

1. Enter the SQL query to retrieve the error message. This SQL query should select only one column. Choose the Test Query button to validate the query syntax.

2. Choose Apply.

3. Oracle E-Business Suite Desktop Integration Framework displays a list of the parameters used in the query.

   - You can optionally enter a unique reference name for the parameter by which subsequent rules can refer to the parameter.

   - Select the data type for the parameter.

   - Specify the source from which the parameter value will be obtained: Environment Variables, Import, Interface Table, or Upload Parameters.

   - Specify the value based on the selected source. If the source is environment variables, enter the variable to use for the value, such as User ID, Application, Responsibility ID, or Language. If the source is Import, enter the interface attribute or reference parameter to use for the value. If the source is Interface Table, enter the table details. If the source is upload parameters, enter the upload parameter to use for the value.

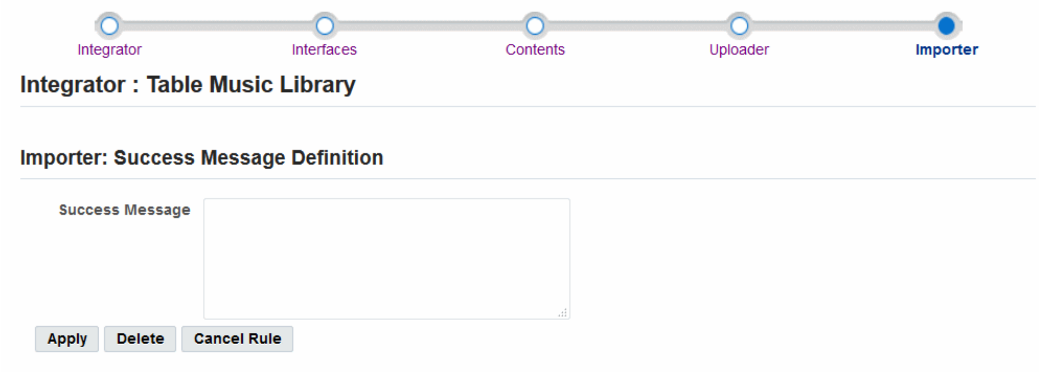
4. Choose Apply.

   Note: If you no longer require this rule definition, you can choose the Delete button to remove it.

To define a success message definition importer rule:

Use a success message definition rule to specify a message to display to the user when the import process is submitted successfully. For an asynchronous concurrent request importer, Oracle Web Applications Desktop Integrator returns the success message after the import program is submitted, without waiting for the program to complete. For a synchronous concurrent request or a PL/SQL API importer, Oracle Web Applications Desktop Integrator returns the success message only after the import process completes and all records have been imported without error.

Importer Page: Success Message Definition Rule



Integrator : Table Music Library

Importer: Success Message Definition

1. Enter the text of the success message.

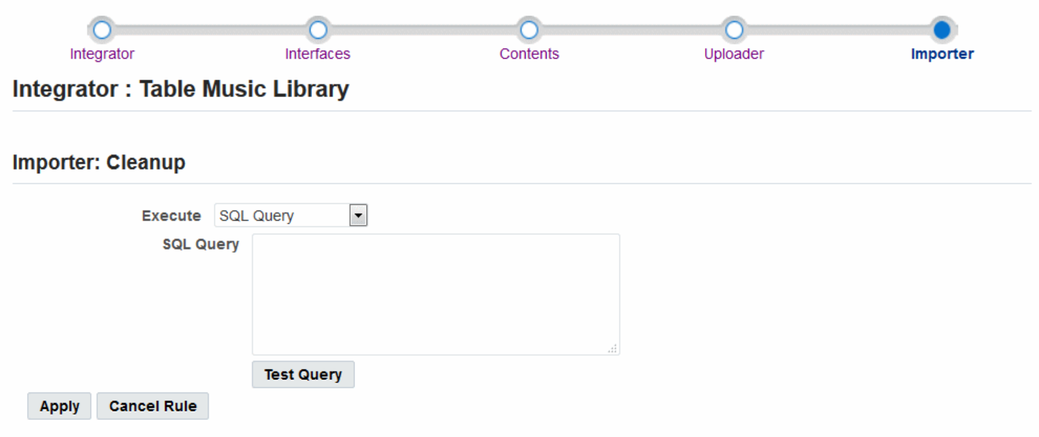
2. Choose Apply.

   Note: If you no longer require this rule definition, you can choose the Delete button to remove it.

To define a cleanup importer rule:

Use a cleanup rule to specify cleanup processing to perform if errors occurred during any of the previous importer rules.

Importer Page: Cleanup Rule



Integrator : Table Music Library

Importer: Cleanup

Execute: SQL Query

SQL Query

1. Select the type of cleanup processing to run: SQL Query, PL/SQL Function, or PL/SQL Procedure.

2. For a SQL query, enter the query. Choose the Test Query button to validate the query syntax.

3. For a PL/SQL function, enter the package name and function name, and specify whether the API returns an error message or an FND message code if it encounters an error.

4. For a PL/SQL procedure, enter the package name and procedure name, and specify whether the API returns an error message or an FND message code if it encounters an error.

5. Choose Apply. Oracle E-Business Suite Desktop Integration Framework displays additional details depending on the processing type.

6. For a SQL query, the page displays a list of the columns selected by the query. You can optionally enter a unique reference name for each column by which subsequent rules can refer to the column.

   The page also displays a list of the parameters used in the query.

   - You can optionally enter a unique reference name for the parameter by which subsequent rules can refer to the parameter.

   - Select the data type for the parameter.

   - Specify the source from which the parameter value will be obtained: Environment Variables, Import, Interface Table, or Upload Parameters.

   - Specify the value based on the selected source. If the source is environment variables, enter the variable to use for the value, such as User ID, Application, Responsibility ID, or Language. If the source is Import, enter the interface attribute or reference parameter to use for the value. If the source is Interface Table, enter the table details. If the source is upload parameters, enter the upload parameter to use for the value.

7. For a PL/SQL function, if the function returns more details about any error that it encounters, select the Returns Error Information option.

8. For a PL/SQL function or procedure, specify whether the API returns an error message or an FND message code if it encounters an error.

9. For a PL/SQL function or procedure, the page displays a list of the parameters for the function or procedure.

   - You can optionally enter a unique reference name for the parameter by which subsequent rules can refer to the parameter.

   - If the parameter is used to retrieve an error message, select the Error Message checkbox.

   - Optionally enter a default value for the parameter.

   - Specify the source from which the parameter value will be obtained: Environment Variables, Import, Interface Table, or Upload Parameters.

   - Specify the value based on the selected source. If the source is environment variables, enter the variable to use for the value, such as User ID, Application, Responsibility ID, or Language. If the source is Import, enter the interface attribute or reference parameter to use for the value. If the source is Interface Table, enter the table details. If the source is upload parameters, enter the upload parameter to use for the value.

   - Optionally specify a parameter list to use for key-value lookups.

   - Optionally enter a SQL clause to modify the value, such as `TO_CHAR()`.

10. Choose Apply.

    Note: If you no longer require this rule definition, you can choose the Delete button to remove it.

3

Managing Components

This chapter covers the following topics:

- Managing Components
- Defining Components

Managing Components

A *component* is a set of metadata that provides a list of values for a field in a desktop document. You can assign a component to an interface attribute to use the list of values defined in the component to validate the value in the document field corresponding to that interface attribute. In this way you can enforce your Oracle E-Business Suite application's business rules for the field values within the desktop document. Use the Components List page to view and manage the components defined for your Oracle E-Business Suite installation.

To manage components:

1. Navigate to the Components List page from the Oracle E-Business Suite Navigator by choosing either Desktop Integration Manager: Manage Components or a custom responsibility and menu path specified by your system administrator.

2. Search for the components you want to display. You can enter the following search criteria:

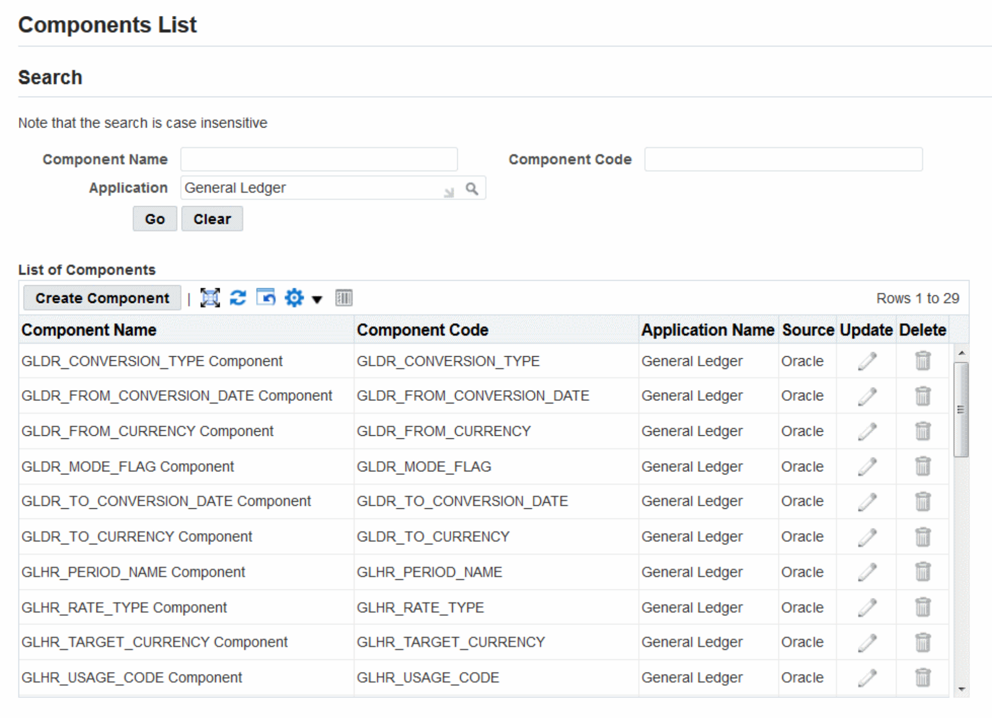
   - Component Name - Enter the display name of the component. You can enter a partial value to search for components whose names contain that value. This field is case-insensitive.

   - Component Code - Enter the unique internal code for the component. You can enter a partial value to search for components whose codes contain that value. This field is case-insensitive.

- Application - Select the application that owns the component.

To perform a search, you must enter a value that does not begin with a percent sign (%) or underscore (_) for at least one search option.

Components List Page



Components List

Search

Note that the search is case insensitive

Component Name		Component Code	
Application	General Ledger		

List of Components

Rows 1 to 29

Component Name	Component Code	Application Name	Source	Update	Delete
GLDR_CONVERSION_TYPE Component	GLDR_CONVERSION_TYPE	General Ledger	Oracle	✎	🗑
GLDR_FROM_CONVERSION_DATE Component	GLDR_FROM_CONVERSION_DATE	General Ledger	Oracle	✎	🗑
GLDR_FROM_CURRENCY Component	GLDR_FROM_CURRENCY	General Ledger	Oracle	✎	🗑
GLDR_MODE_FLAG Component	GLDR_MODE_FLAG	General Ledger	Oracle	✎	🗑
GLDR_TO_CONVERSION_DATE Component	GLDR_TO_CONVERSION_DATE	General Ledger	Oracle	✎	🗑
GLDR_TO_CURRENCY Component	GLDR_TO_CURRENCY	General Ledger	Oracle	✎	🗑
GLHR_PERIOD_NAME Component	GLHR_PERIOD_NAME	General Ledger	Oracle	✎	🗑
GLHR_RATE_TYPE Component	GLHR_RATE_TYPE	General Ledger	Oracle	✎	🗑
GLHR_TARGET_CURRENCY Component	GLHR_TARGET_CURRENCY	General Ledger	Oracle	✎	🗑
GLHR_USAGE_CODE Component	GLHR_USAGE_CODE	General Ledger	Oracle	✎	🗑

3. To review a component definition or to update the parameter values for an advanced calendar, calendar, Java-validated, or table-validated component, choose the update icon for that component. See: Defining Components, page 3-3.

    Note: You can only update custom components.

4. To delete a component, choose the delete icon for that component.

    Note: You can only delete custom components.

5. To define a new component, select the Create Component button. See: Defining Components, page 3-3.

Defining Components

Use the Component Definition page to create a new component or to update an existing component.

> **Note:** You can only update custom components.

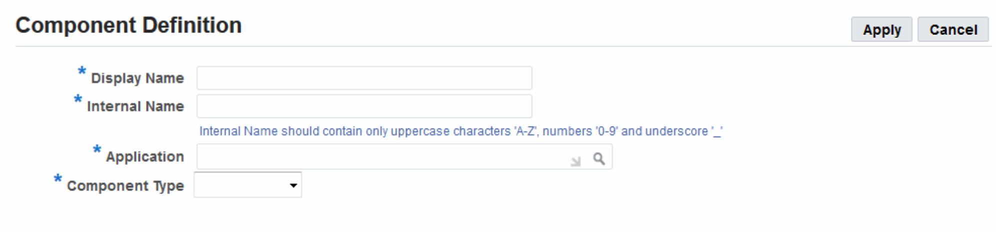
To define a component:

1. Navigate to the Component Definition page by choosing the Create Component button or the update icon in the Components List page.

   If you are updating an existing component, then the component properties are populated with previously defined information for the selected component.

   > **Note:** To preserve consistency, you cannot update the basic properties of an existing component. You can only update the values for the component parameters.

   Component Definition Page

   Internal Name should contain only uppercase characters 'A-Z', numbers '0-9' and underscore '_'

2. Enter the display name of the component.

3. Enter a unique internal name, or component code, for the component. The internal name can include only the following characters:

   - Uppercase letters from `A` through `Z`

   - Numerals from `0` through `9`

   - The underscore character (_)

4. Select the application that owns the integrator.

5. Select the component type.

   - Advanced Calendar - An advanced calendar component that supports both a date and time picker for the Gregorian, English Hijrah, Arabic Hijrah, and Thai Buddhist calendars.

For an advanced calendar component, the page displays the predefined Java class `oracle.apps.bne.integrator.component.BneDateTimeCalendarComponent` as the component entity.

- Calendar - A calendar component that supports only a date picker for the Gregorian calendar.

  For a calendar component, the page displays the predefined Java class `oracle.apps.bne.integrator.component.BneCalendarComponent` as the component entity.

- Flex - A flexfield component.

  For a flexfield component, select the flexfield parameter list.

- JAVA LOV - A Java-validated component.

  For a Java-validated component, enter the fully qualified class name of the Java class to use as the component entity.

- TABLE LOV - A table-validated component.

  For a table-validated component, the page displays the predefined Java class `BneOAValueSet` as the component entity.

6. Choose Apply.

7. For an advanced calendar, calendar, Java-validated, or table-validated component, Oracle E-Business Suite Desktop Integration Framework displays a list of the parameters for the component. Specify values for the component parameters.

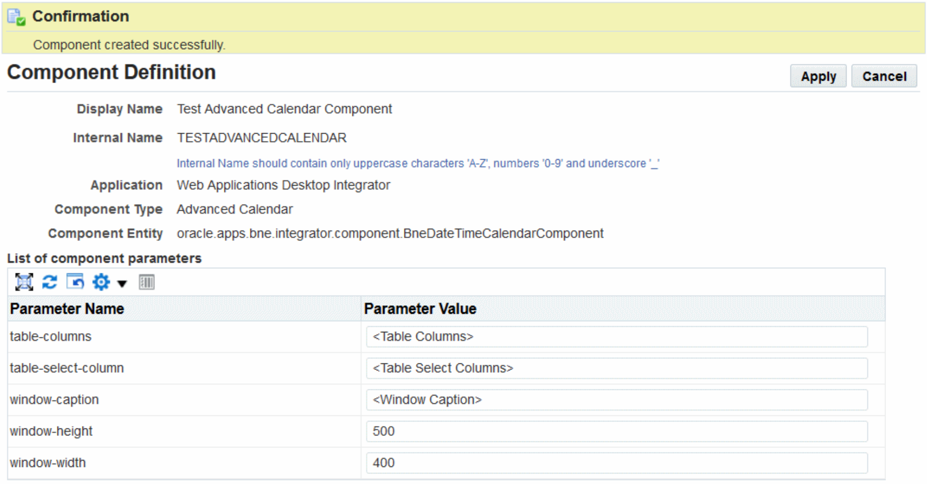
   - For an advanced calendar component, specify the following parameters.

     - window-height - The height of the date picker window. The default value is 500.

     - window-width - The width of the date picker window. The default value is 400.

       > **Note:** Do not specify values for the table-columns, table-select-column, and window-caption parameters. These parameters are not used for advanced calendar components.

Component Definition Page with List of Component Parameters for an Advanced Calendar Component



Confirmation

Component created successfully.

Component Definition

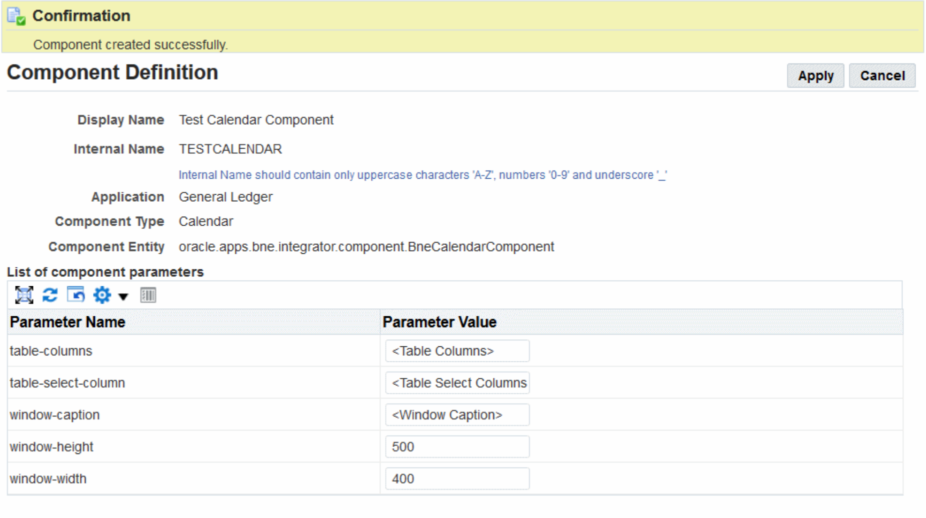
Display Name	Test Advanced Calendar Component
Internal Name	TESTADVANCEDCALENDAR Internal Name should contain only uppercase characters 'A-Z', numbers '0-9' and underscore '_'
Application	Web Applications Desktop Integrator
Component Type	Advanced Calendar
Component Entity	oracle.apps.bne.integrator.component.BneDateTimeCalendarComponent

List of component parameters

Parameter Name	Parameter Value
table-columns	<Table Columns>
table-select-column	<Table Select Columns>
window-caption	<Window Caption>
window-height	500
window-width	400

- For a calendar component, specify the following parameters.
  - table-columns - A list of the columns to display in the window, separated by commas.
  - table-select-column - The name of the interface column that the date picker updates in the desktop document.
  - window-caption - The caption to display in the date picker window.
  - window-height - The height of the date picker window. The default value is 500.
  - window-width - The width of the date picker window. The default value is 400.

Component Definition Page with List of Component Parameters for a Calendar Component



Confirmation

Component created successfully.

Component Definition

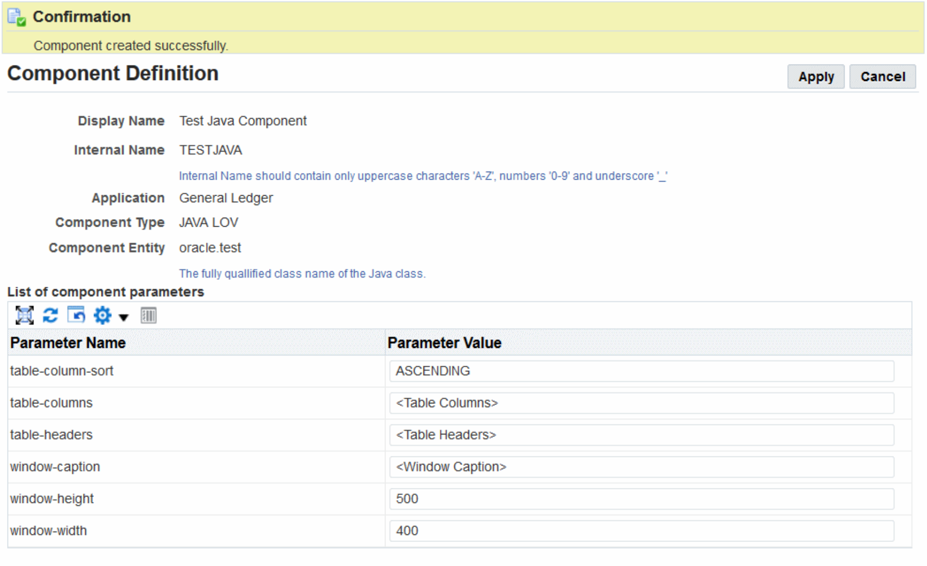
Display Name	Test Calendar Component
Internal Name	TESTCALENDAR Internal Name should contain only uppercase characters 'A-Z', numbers '0-9' and underscore '_'
Application	General Ledger
Component Type	Calendar
Component Entity	oracle.apps.bne.integrator.component.BneCalendarComponent

List of component parameters

Parameter Name	Parameter Value
table-columns	<Table Columns>
table-select-column	<Table Select Columns
window-caption	<Window Caption>
window-height	500
window-width	400

- For a Java-validated component, specify the following parameters.
  - table-column-sort - The sort order for the table in the list of values window. The default value is `ASCENDING`.
  - table-columns - A list of the columns to display in the list of values window, separated by commas.
  - table-headers - A list of the column headers for the columns that will appear in the list of values window, separated by commas.
  - window-caption - The caption to display in the list of values window.
  - window-height - The height of the list of values window. The default value is 500.
  - window-width - The width of the list of values window. The default value is 400.

Component Definition Page with List of Component Parameters for a Java-Validated Component



Confirmation

Component created successfully.

Component Definition

Display Name	Test Java Component
Internal Name	TESTJAVA Internal Name should contain only uppercase characters 'A-Z', numbers '0-9' and underscore '_'
Application	General Ledger
Component Type	JAVA LOV
Component Entity	oracle.test The fully qualified class name of the Java class.

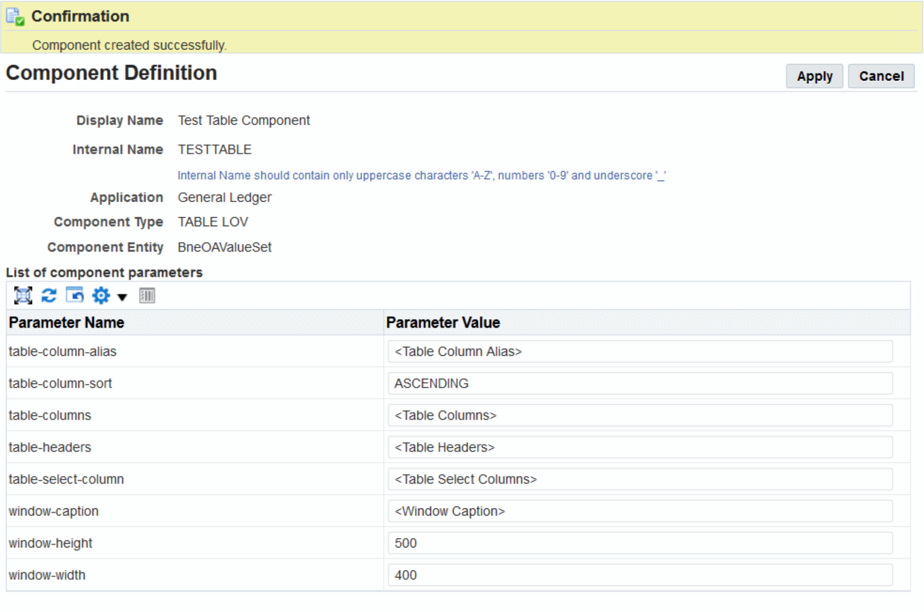
List of component parameters

Parameter Name	Parameter Value
table-column-sort	ASCENDING
table-columns	<Table Columns>
table-headers	<Table Headers>
window-caption	<Window Caption>
window-height	500
window-width	400

- For a table-validated component, specify the following parameters.
  - table-column-alias - If the interface column that the list of values updates is not among the columns displayed in the list of values window, then enter the interface column in this parameter to create an alias for the displayed column that you want to insert into the desktop document.
  - table-column-sort - The sort order for the table in the list of values window. The default value is `ASCENDING`.
  - table-columns - A list of the columns to display in the list of values window, separated by commas.
  - table-headers - A list of the column headers for the columns that will appear in the list of values window, separated by commas.
  - table-select-column - The name of the interface column that the list of values updates in the desktop document.
  - window-caption - The caption to display in the list of values window.
  - window-height - The height of the list of values window. The default value is 500.
  - window-width - The width of the list of values window. The default value

is 400.

Component Definition Page with List of Component Parameters for a Table-Validated Component



Confirmation

Component created successfully.

Component Definition

Display Name	Test Table Component
Internal Name	TESTTABLE Internal Name should contain only uppercase characters 'A-Z', numbers '0-9' and underscore '_'
Application	General Ledger
Component Type	TABLE LOV
Component Entity	BneOAValueSet

List of component parameters

Parameter Name	Parameter Value
table-column-alias	<Table Column Alias>
table-column-sort	ASCENDING
table-columns	<Table Columns>
table-headers	<Table Headers>
table-select-column	<Table Select Columns>
window-caption	<Window Caption>
window-height	500
window-width	400

8. Choose Apply.

4

Managing Parameters

This chapter covers the following topics:

- Managing Parameters

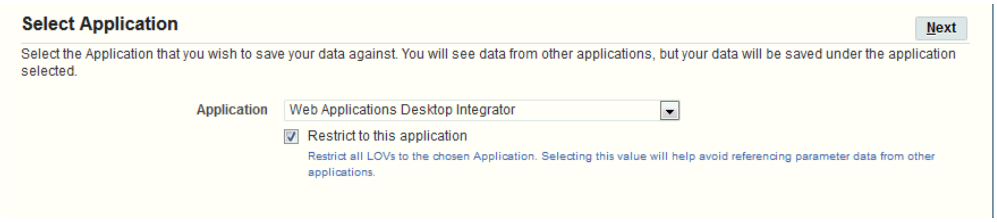
Managing Parameters

Use the parameter list definition pages to define and manage parameter lists and parameters.

To manage parameter lists and parameters:

1. Navigate to the parameter list definition pages from the Oracle E-Business Suite Navigator by choosing either Desktop Integration Manager: Define Parameter or a custom responsibility and menu path specified by your system administrator.

Select Application Page



Select Application

Select the Application that you wish to save your data against. You will see data from other applications, but your data will be saved under the application selected.

Application: Web Applications Desktop Integrator

☑ Restrict to this application

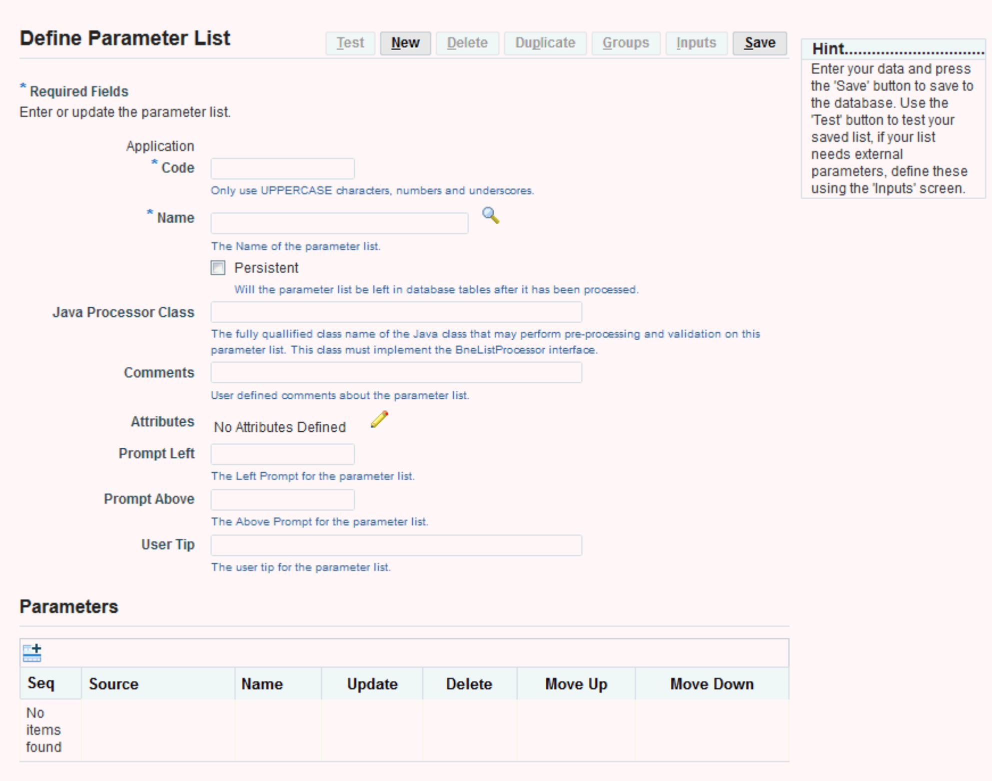
Restrict all LOVs to the chosen Application. Selecting this value will help avoid referencing parameter data from other applications.

2. Select the application that owns the parameter list, against which the parameter data will be saved.

3. To restrict all the lists of values in the subsequent pages to display only values from the selected application, select the Restrict to this Application option. Selecting this option helps ensure that you do not reference data from other applications in this parameter list if you want to maintain the list solely in a single application.

4. Select Next.

Define Parameter List Page



Define Parameter List

Hint.............................
Enter your data and press the 'Save' button to save to the database. Use the 'Test' button to test your saved list, if your list needs external parameters, define these using the 'Inputs' screen.

* Required Fields

Enter or update the parameter list.

Application * Code

Only use UPPERCASE characters, numbers and underscores.

* Name

The Name of the parameter list.

□ Persistent

Will the parameter list be left in database tables after it has been processed.

Java Processor Class

The fully qualified class name of the Java class that may perform pre-processing and validation on this parameter list. This class must implement the BneListProcessor interface.

Comments

User defined comments about the parameter list.

Attributes: No Attributes Defined

Prompt Left

The Left Prompt for the parameter list.

Prompt Above

The Above Prompt for the parameter list.

User Tip

The user tip for the parameter list.

Parameters

Seq	Source	Name	Update	Delete	Move Up	Move Down
No items found						

5. In the Define Parameter List page you can create a new parameter list.

   - If no parameter list is currently displayed, enter the information for the new parameter list in the blank fields and select Save.

   - If another parameter list is displayed, first select New to clear the page. Then enter information for the new parameter list and select Save.

   See: To define a parameter list, page 4-3.

6. To view or update an existing parameter list, select the display name of the parameter list you want in the Name field. See: To define a parameter list, page 4-3.

7. To define a new parameter list by copying the definition of an existing list, first display the definition of the existing list and then select Duplicate. Enter the name, application, and unique code for the new list, and select Save.

8. To define parameter groups for the currently displayed parameter list, select Groups. See: To define parameter groups, page 4-11.

9. If the currently displayed parameter list requires external API parameters, select Inputs to enter test data for those parameters before testing the parameter list. See: To define inputs, page 4-13.

10. To test how the currently displayed parameter list will appear, select Test. See: To test a parameter list, page 4-13.

11. To delete the currently displayed parameter list, select Delete.

To define a parameter list:

1. In the Define Parameter List page, to create a new list, enter a unique code to identify the parameter list. The code can include only the following characters:

   - Uppercase letters from `A` through `Z`

   - Numerals from `0` through `9`

   - The underscore character (_)

   > **Note:** To preserve consistency, you cannot update the application or code for an existing parameter list.

2. In the Name field, enter the display name of the parameter list.

3. If you want the parameter list to remain in the Oracle Web Applications Desktop Integrator database tables after the list has been processed, select the Persistent option. If you do not select this option, the parameter list is deleted from the database tables after the list has been processed.

4. Optionally enter the fully qualified class name of a Java class that can perform pre-processing and validation on this parameter list. This class must implement the BneListProcessor interface.

5. Optionally enter comments about the parameter list.

6. The Define Parameter List page indicates whether any attributes are defined for the parameter list. To create or update attributes for the parameter list, select the Update List Attributes icon. See: To define attributes, page 4-4.

7. Optionally enter a prompt to appear to the left of the parameter list if it appears in the header region of the document.

8. Optionally enter a prompt to appear above the parameter list if it appears in the lines region of the document.

9. Optionally enter a tip to provide users further guidance on the parameter list.

10. Select Save.

11. In the Parameters region, select Add Another Row to add a new parameter to the parameter list. See: To define a parameter, page 4-6.

12. To update a parameter definition, choose the update icon for that parameter. See: To define a parameter, page 4-6.

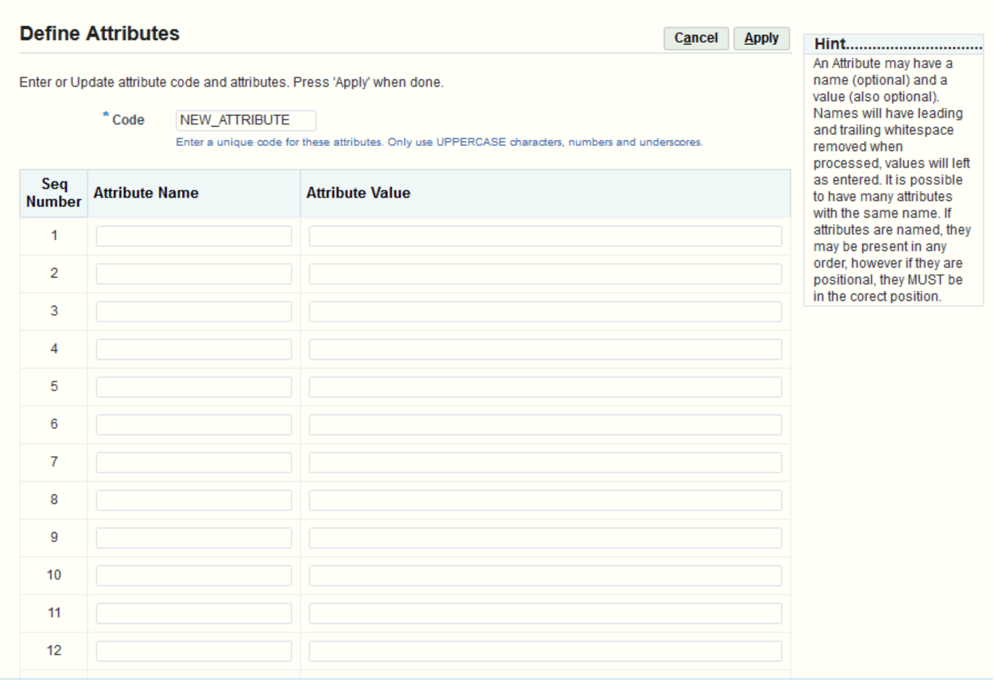
13. To delete a parameter from the list, choose the delete icon for that parameter.

14. To change the position of a parameter in the sequence, choose the move up icon or the move down icon for that parameter.

To define attributes:

1. Use the Define Attributes page to define attributes for a parameter list, a parameter, or a parameter group.

Define Attributes Page



Define Attributes

Enter or Update attribute code and attributes. Press 'Apply' when done.

* Code: NEW_ATTRIBUTE

Enter a unique code for these attributes. Only use UPPERCASE characters, numbers and underscores.

Seq Number	Attribute Name	Attribute Value
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		

Hint.............................

An Attribute may have a name (optional) and a value (also optional). Names will have leading and trailing whitespace removed when processed, values will left as entered. It is possible to have many attributes with the same name. If attributes are named, they may be present in any order, however if they are positional, they MUST be in the corect position.

Enter a unique code to identify the attributes. The code can include only the following characters:

- Uppercase letters from `A` through `Z`
- Numerals from `0` through `9`
- The underscore character (_)

2. To define an attribute, enter an attribute name, an attribute value, or both. You can define multiple attributes with the same name. If you define named attributes, then you can enter the attributes in any order. However, if you define positional attributes, meaning you only enter attribute values and do not provide attribute names, then you must define the attributes in the correct position.

   You can define up to 30 attributes.

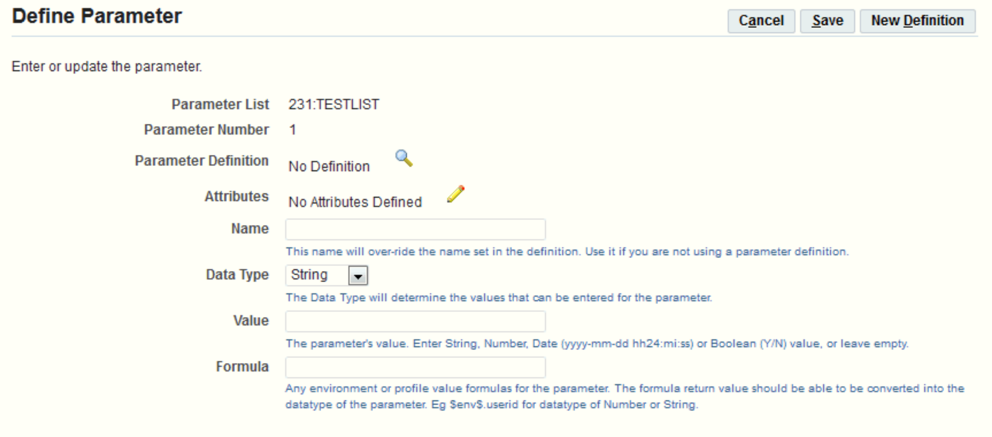
   > **Note:** Oracle E-Business Suite Desktop Integration Framework removes leading and trailing spaces from attribute names when processing the names. However, attribute values are preserved exactly as you enter them.

3. Select Apply.

To define a parameter:

1. The Define Parameter page displays the parameter list to which this parameter belongs and the sequence number for the parameter.

   Define Parameter Page

   Define Parameter

   Enter or update the parameter.

Parameter List	231:TESTLIST
Parameter Number	1
Parameter Definition	No Definition
Attributes	No Attributes Defined
Name	This name will over-ride the name set in the definition. Use it if you are not using a parameter definition.
Data Type	String The Data Type will determine the values that can be entered for the parameter.
Value	The parameter's value. Enter String, Number, Date (yyyy-mm-dd hh24:mi:ss) or Boolean (Y/N) value, or leave empty.
Formula	Any environment or profile value formulas for the parameter. The formula return value should be able to be converted into the datatype of the parameter. Eg env.userid for datatype of Number or String.

   To base the parameter you are adding on an existing parameter metadata definition, select the parameter definition you want.

   - To create a new parameter metadata definition, select New Definition.

   - To update an existing parameter metadata definition, select the parameter definition you want and then choose the Update Parameter Definition icon.

   See: To define parameter metadata, page 4-7.

2. The Define Parameter page indicates whether any attributes are defined for the parameter. To create or update attributes for the parameter, select the Update Parameter Attributes icon. See: To define attributes, page 4-4.

3. Enter a name for the parameter. If you are using an existing parameter definition, the name you enter here overwrites the previous name for the definition. Usually you should only enter a name when you are creating a new parameter not based on an existing definition.

4. Select the data type for the parameter.

   The data type can be String, Number, Date, or Boolean.

5. Optionally enter a default parameter value.

   - For a date parameter, enter the value in the following format: `yyyy-mm-dd hh24:mi:ss`

- For a Boolean parameter, enter either `Y` or `N`.

  If you enter a value, you must also enter a description for the parameter. If you do not enter a parameter value, then the description is optional.

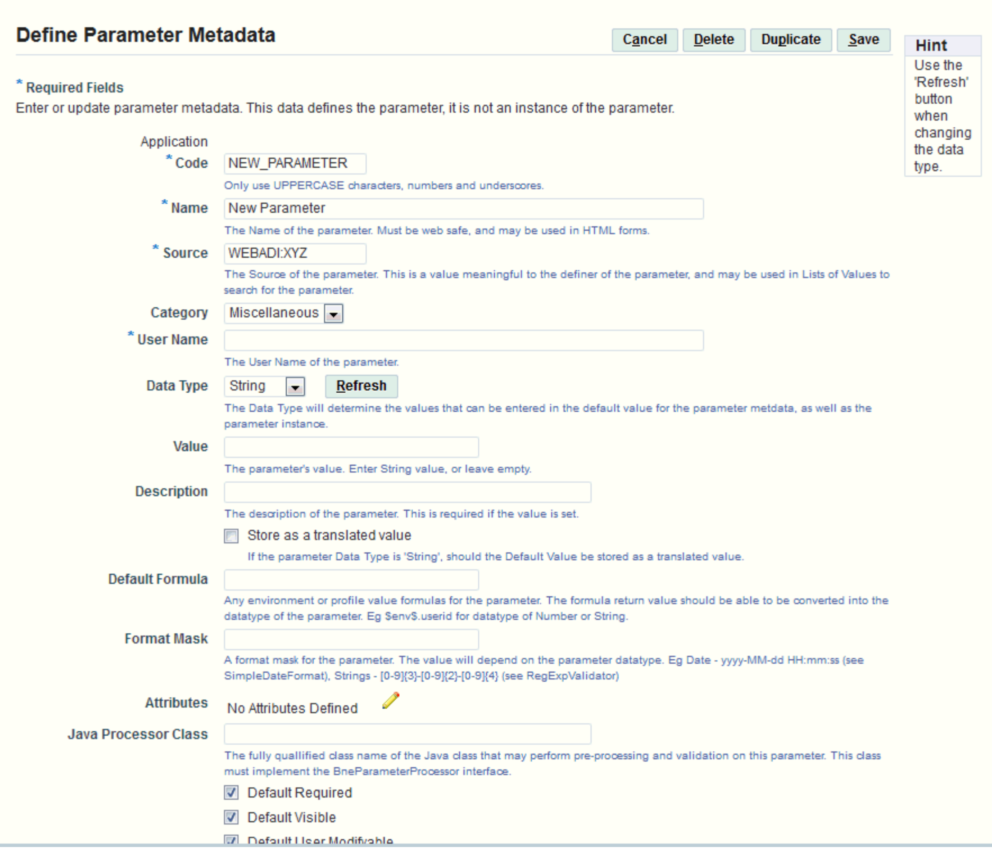
6. Optionally enter any environment variable or profile option value formulas to derive the default value for the parameter. The formula should return a value that can be converted into the data type of the parameter. For example, a possible formula for a parameter with a data type of Number or String could be: `$env$.userid`

7. Select Save.

To define parameter metadata:

1. The Define Parameter Metadata page displays the application to which the parameter definition belongs.

Define Parameter Metadata Page



Define Parameter Metadata

Hint

Use the 'Refresh' button when changing the data type.

* Required Fields

Enter or update parameter metadata. This data defines the parameter, it is not an instance of the parameter.

Application

* Code: NEW_PARAMETER

Only use UPPERCASE characters, numbers and underscores.

* Name: New Parameter

The Name of the parameter. Must be web safe, and may be used in HTML forms.

* Source: WEBADI:XYZ

The Source of the parameter. This is a value meaningful to the definer of the parameter, and may be used in Lists of Values to search for the parameter.

Category: Miscellaneous

* User Name:

The User Name of the parameter.

Data Type: String

The Data Type will determine the values that can be entered in the default value for the parameter metdata, as well as the parameter instance.

Value:

The parameter's value. Enter String value, or leave empty.

Description:

The description of the parameter. This is required if the value is set.

☐ Store as a translated value

If the parameter Data Type is 'String', should the Default Value be stored as a translated value.

Default Formula:

Any environment or profile value formulas for the parameter. The formula return value should be able to be converted into the datatype of the parameter. Eg env.userid for datatype of Number or String.

Format Mask:

A format mask for the parameter. The value will depend on the parameter datatype. Eg Date - yyyy-MM-dd HH:mm:ss (see SimpleDateFormat), Strings - [0-9]{3}-[0-9]{2}-[0-9]{4} (see RegExpValidator)

Attributes: No Attributes Defined

Java Processor Class:

The fully quallified class name of the Java class that may perform pre-processing and validation on this parameter. This class must implement the BneParameterProcessor interface.

☑ Default Required

☑ Default Visible

☑ Default User Modifiable

Enter a unique code to identify the parameter definition. The code can include only the following characters:

- Uppercase letters from A through Z
- Numerals from 0 through 9
- The underscore character (_)

If you are viewing an existing parameter definition, you can switch to view a different parameter definition by searching for and selecting the code for that definition.

2. Enter a display name for the parameter. The display name must be web safe and must be in a format that is suitable for display in HTML forms.

3. Enter a source for the parameter. You can use the source to search for this parameter definition in lists of values in other Oracle E-Business Suite Desktop Integration Framework pages.

4. Select the category to which the parameter belongs.

   - Miscellaneous
   - Position
   - Appearance
   - Behavior
   - Data
   - Field

5. Enter a user name for the parameter.

6. Select the data type for the parameter: String, Number, Date, Boolean, or Key Flex. Then choose Refresh to update the subsequent fields based on the selected data type.

7. For a key flexfield parameter, select the application to which the flexfield belongs, enter the flexfield structure code, and enter the flexfield number.

8. Optionally enter a default parameter value.

   - For a date parameter, enter the value in the following format: `yyyy-mm-dd hh24:mi:ss`
   - For a Boolean parameter, select either None, True, or False.
   - For a key flexfield parameter, enter the concatenated flexfield value.

   If you enter a value, you must also enter a description for the parameter. If you do not enter a parameter value, then the description is optional.

9. For a string parameter, if you want to store the default value as a translated value, select the Store as a Translated Value option.

10. Optionally enter any environment variable or profile option value formulas to derive the default value for the parameter. The formula should return a value that can be converted into the data type of the parameter. For example, a possible formula for a parameter with a data type of Number or String could be: `$env$.userid`

11. Optionally enter a format mask for the parameter. The value for the format mask depends on the parameter data type. For example, a format mask for a date parameter could be: `yyyy-MM-dd HH:mm:ss`

    An example format mask for a string parameter could be: `[0-9]{3}-[0-9]{2}`

```
-[0-9]{4}
```

12. The Define Parameter Metadata page indicates whether any attributes are defined for the parameter. To create or update attributes for the parameter, select the Update Parameter Attributes icon. See: To define attributes, page 4-4.

13. Optionally enter the fully qualified class name of a Java class that can perform pre-processing and validation on this parameter. This class must implement the BneParameterProcessor interface.

14. Select whether a default value is required for the parameter.

15. Select whether the default value for the parameter is visible to users.

16. Select whether users can modify the parameter value to a value other than the default.

17. Select the validation type for the parameter: None, Fnd Lookups, Value Set, or Bne Query.

18. Depending on the selected validation type, enter the lookup type, value set, or query against which you want to validate the parameter value. If you selected the validation type None, leave the Validation Value field blank.

19. Enter the maximum size for the parameter value. The default value is 100. The highest possible value you can specify for the maximum size is 99999.

20. Enter the display size for the parameter value. The display size must be equal to or less than the maximum size. The default value is 100.

21. Select the display type for the parameter.

    - For a string parameter, you can select List Box, CheckBox, Radio Buttons, Text Field, or Password Field.

    - For a number or date parameter, you can select List Box, Radio Buttons, Text Field, or Password Field.

    - For a Boolean parameter, you can select List Box, CheckBox, Radio Buttons, or Text Field.

    - A key flexfield parameter must use the Text Field display type.

22. If you selected the display type Radio Buttons, select whether the display style of the radio buttons should be vertical or horizontal. For all other display types, leave the display style set to None.

23. Optionally enter a prompt to appear to the left of the parameter if it appears in the

header region of the document.

24. Optionally enter a prompt to appear above the parameter if it appears in the lines region of the document.

25. Optionally enter a tip to provide users further guidance on the parameter.

26. Optionally enter a key used to access this parameter on a page. The access key is typically used with the <ALT> key.

27. Optionally enter a URL that the application can call to provide users additional help on this parameter.

28. Select Save.

29. To create a new parameter definition by copying the currently displayed definition, select Duplicate. Enter the user name, application, and unique code for the new parameter definition, and select Save.

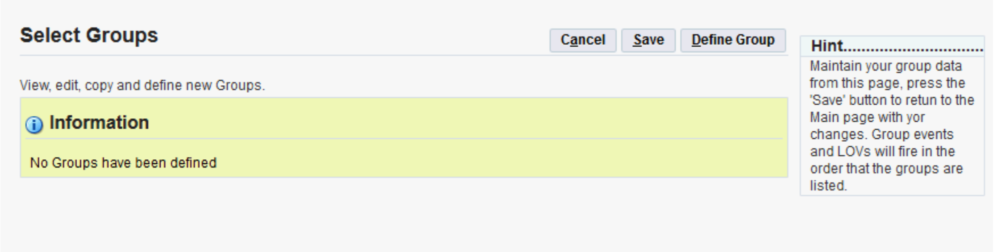
30. To delete the currently displayed parameter definition, select Delete.

To define parameter groups:

If validation of one parameter depends on the value of another parameter, or if several parameters are validated the same way, you can associate those parameters with each other by defining a parameter group and perform validation for all parameters in the group together. You can still also perform validation on each parameter individually if necessary.

1. The Select Groups page displays any parameter groups defined for the parameter list.

Select Groups Page



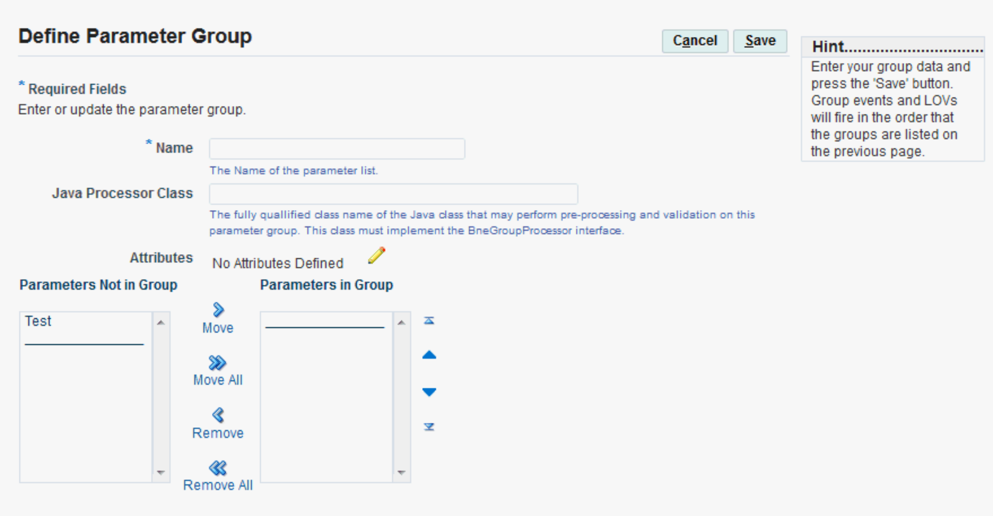
View, edit, copy and define new Groups.

No Groups have been defined

Maintain your group data from this page, press the 'Save' button to retun to the Main page with yor changes. Group events and LOVs will fire in the order that the groups are listed.

To define a new parameter group, select Define Group.

Define Parameter Group Page



Define Parameter Group

* Required Fields

Enter or update the parameter group.

* Name

The Name of the parameter list.

Java Processor Class

The fully qualified class name of the Java class that may perform pre-processing and validation on this parameter group. This class must implement the BneGroupProcessor interface.

Attributes: No Attributes Defined

Parameters Not in Group	Parameters in Group
Test	

Hint...

Enter your group data and press the 'Save' button. Group events and LOVs will fire in the order that the groups are listed on the previous page.

2. In the Define Parameter Group page, enter a name for the parameter group.

3. Optionally enter the fully qualified class name of a Java class that can perform pre-processing and validation on this parameter group. This class must implement the BneGroupProcessor interface.

4. The Define Parameter Group page indicates whether any attributes are defined for the parameter group. To create or update attributes for the parameter group, select the Update Group Attributes icon. See: To define attributes, page 4-4.

5. Select the parameters you want in the Parameters Not in Group list and move them to the Parameters in Group list.

6. Select Save.

7. In the Select Groups page, to update a parameter group, choose the update icon for that group.

8. To define a new parameter group by copying the definition of an existing group, choose the duplicate icon for that group. Enter a unique name for the new group, and select Save.

9. To delete a parameter group from the list, choose the delete icon for that group.

10. To change the position of a parameter group in the sequence, choose the move up icon or the move down icon for that group.

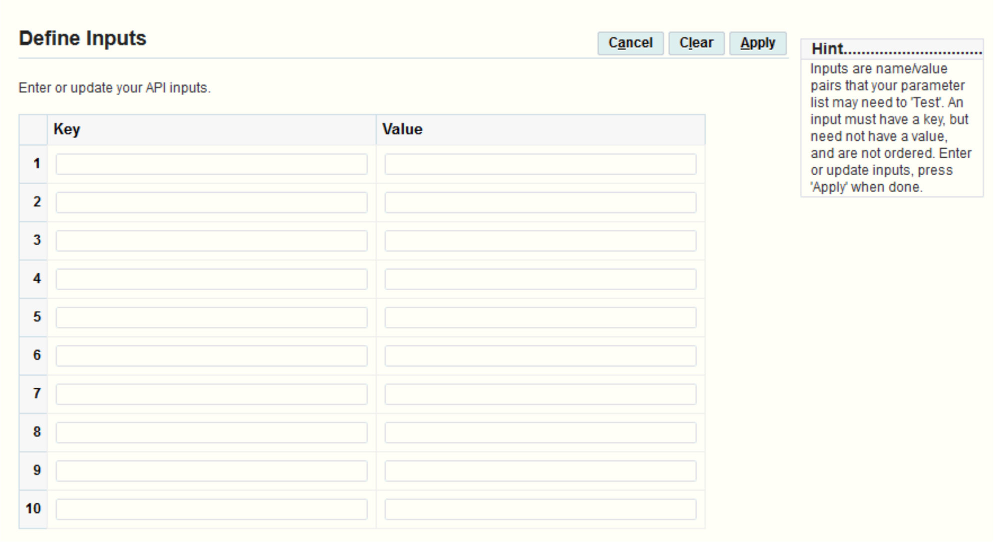
    Oracle Web Applications Desktop Integrator will run group events and lists of values in the sequence in which the parameter groups are listed here.

11. Select Save.

To define inputs:

1. Use the Define Inputs page to provide test data for any external API inputs that a parameter list requires before you test how the parameter list will be rendered.

Define Inputs Page



Define Inputs

Enter or update your API inputs.

	Key	Value
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

Hint

Inputs are name/value pairs that your parameter list may need to 'Test'. An input must have a key, but need not have a value, and are not ordered. Enter or update inputs, press 'Apply' when done.

For each input, enter the key and optionally enter a test value.

You can enter the inputs in any order. You can enter up to 10 inputs.

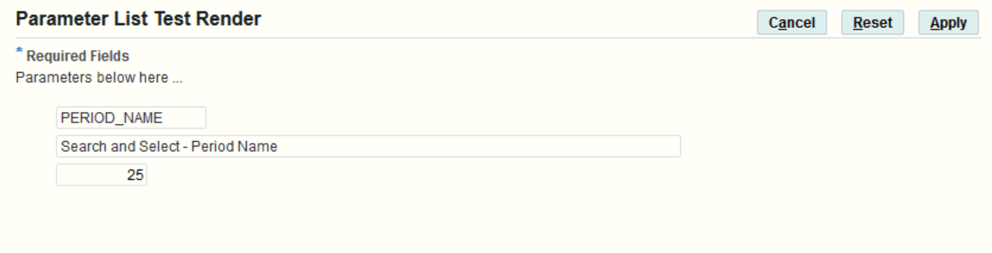
2. To delete all inputs and clear the list, select Clear.

3. To save the list of inputs, select Apply.

To test a parameter list:

1. The Parameter List Test Render page shows how the parameters in the list will be rendered in the desktop application.

Parameter List Test Render Page


Parameter List Test Render

* Required Fields

Parameters below here ...

PERIOD_NAME

Search and Select - Period Name

25

Enter valid test values in the parameter fields and choose Apply.

- If the validation succeeds according to the parameter list definition, then Oracle E-Business Suite Desktop Integration Framework displays the Define Parameter List page again.
- If there are any errors in the validation or in the parameter list definition, then Oracle E-Business Suite Desktop Integration Framework displays an error message.

2. In the Parameter List Test Render page, choose Reset to clear any test values and reset the parameter fields to their default values. You can then enter new values and choose Apply to test again.
3. If necessary, update the parameter definitions to correct any errors.

To define a parameter list to use a WHERE clause with a key flexfield component:

If you define an interface attribute that uses key flexfield validation, and you want to use a WHERE clause to restrict the code combinations included in the list of values, you can define a parameter list for the flexfield component that includes a parameter named `field:whereClause`. The WHERE clause also prevents a user from entering a combination that does not fit the WHERE clause.

> **Note:** You should not use a WHERE clause for a key flexfield that allows dynamic inserts.

See: Flexfield Definition Arguments, *Oracle E-Business Suite Developer's Guide*.

1. Define a parameter list that you will associate with the key flexfield component for the interface attribute.
2. Define a parameter named `field:whereClause`. This parameter must use the string data type. Enter the WHERE clause as the parameter value. Do not specify the word "WHERE" in this value.

You can use tokens to reference one or more other interface attributes within the WHERE clause. Use the following syntax for the token for a referenced attribute:

```
$Table$.<INTERFACE_NAME>.<INTERFACE_ATTRIBUTE>
```

Replace *<INTERFACE_NAME>* with the name of the interface and *<INTERFACE_ATTRIBUTE>* with the name of the referenced attribute.

For example, the value of a `field:whereClause` parameter that restricts code combinations based on a subinventory code could be defined as follows:

```
SUBINVENTORY_CODE = $Table$.EXAMPLE_KFF_INTERFACE.SUBINVENTORY_CODE
```

3. Define an additional parameter in the parameter list for each interface attribute referenced by a token in the WHERE clause. Use the following syntax for the name of each such parameter:

   ```
   <INTERFACE_NAME>.<INTERFACE_ATTRIBUTE>
   ```

   Replace *<INTERFACE_NAME>* with the name of the interface and *<INTERFACE_ATTRIBUTE>* with the name of the referenced attribute.

   These parameters must use the string data type. Leave the value for these parameters blank.

4. Define a parameter group to group together the parameters for the interface attributes that the WHERE clause references. The parameter group should also be named `field:whereClause`.

5. Associate the parameter list with a component that uses flexfield validation. See: Defining Components, page 3-3.

6. Associate the flexfield component with the interface attribute that uses the key flexfield for its validation. See: To define an interface attribute, page 2-16.

To define a parameter list to use a data set identifier with a key flexfield component:

If you define an interface attribute that uses key flexfield validation, and you want to use a data set identifier to specify which set of code combinations to use for this flexfield, you can define a parameter list for the flexfield component that includes a parameter named `field:datasetCol`. See: Flexfield Definition Arguments, *Oracle E-Business Suite Developer's Guide*.

1. Define a parameter list that you will associate with the key flexfield component for the interface attribute.

2. Define a parameter named `field:datasetCol`. This parameter must use the string data type. Enter the interface attribute that contains the data set identifier as the parameter value. Use the following syntax to specify the value:

   ```
   <INTERFACE_NAME>.<INTERFACE_ATTRIBUTE>
   ```

Replace <*INTERFACE_NAME*> with the name of the interface and <*INTERFACE_ATTRIBUTE*> with the name of the attribute that contains the data set identifier.

For example, the value of a `field:datasetCol` parameter with a data set identified by the organization ID could be defined as follows:

```
EXAMPLE_KFF_INTERFACE.ORGANIZATION_ID
```

3. Associate the parameter list with a component that uses flexfield validation. See: Defining Components, page 3-3.

4. Associate the flexfield component with the interface attribute that uses the key flexfield for its validation. See: To define an interface attribute, page 2-16.

To define a parameter list to specify handling for a text file content:

For a text file content, you can optionally define parameters that let you specify default values for the character to use as the file delimiter, whether to ignore consecutive delimiters, and the line at which to start importing. If you choose to use these parameters, you must include them in the parameter list associated with the text file content.

> **Note:** Users can override the default values for these parameters when creating a document. See: Creating a Spreadsheet, *Oracle Web Applications Desktop Integrator Implementation and Administration Guide*.

1. Define a parameter list that you will associate with the text file content.

2. To specify the character to use as the text file delimiter, define a parameter named `bnectl:fileDelimiter`. This parameter must use the string data type. Set the default parameter value to the character you want to use as the delimiter.

3. To specify whether to ignore consecutive delimiters, define a parameter named `bnectl:ignoreConsecutiveDelim`. This parameter must use the Boolean data type. Set the default parameter value to `Y` to ignore consecutive delimiters, or `N` to insert a null value into the spreadsheet when two delimiters exist side by side.

4. To specify the line at which to start importing, define a parameter named `bnectl:lineToImportFrom`. This parameter must use the number data type. Set the default parameter value to the number of the line within the text file at which the import should start. For example, if the text file includes header lines before the data lines, you can set this parameter to the line number of the first data line.

5. Associate the parameter list with the text file content. See: To define contents, page 2-27.

To define a document parameter list:

You can optionally define parameters to reference runtime data that you want to include when a document is created. The list that contains these parameters is known as the document parameter list for the integrator. You can reference parameters from this list when you define a layout for the integrator.

1. Define a parameter list that you will specify as the document parameter list for the integrator.

2. Define the runtime parameters that you want to reference in a layout. The parameter names must follow these guidelines:

   - Each parameter name must begin with the following string:

     `bne:custom:`

   - The next character after this string must be a letter.

   - The parameter name must not contain any special characters except the underscore character (_) and the colon character (`:`).

3. Select this parameter list as the document parameter list for the integrator in the Integrator Parameters region of the Integrator Information page. See: To define integrator information, page 2-5.

4. When defining a layout for this integrator, set the default type for a layout field to `Parameter`, and set the default value for that field to the runtime parameter name you defined in step 2. See: Defining Layouts, *Oracle Web Applications Desktop Integrator Implementation and Administration Guide*.

A

Loading Integrator Definitions

This appendix covers the following topics:

- Loading Integrator Definitions

Loading Integrator Definitions

You can use the Generic Loader (FNDLOAD) concurrent program to download integrator definitions from an Oracle E-Business Suite instance to a text file, and to upload integrator definitions from a text file to an Oracle E-Business Suite instance. For example, you can download an integrator definition from a development or test instance and upload it to a production instance. You can also download and store a text file representation of an integrator definition for source control purposes.

Use the `bneintegrator.lct` configuration file with the Generic Loader to load integrator definitions. The `bneintegrator.lct` file is located in the `$BNE_TOP/patch/115/import/` directory.

The entity for this configuration file is `BNE_INTEGRATORS`. This entity includes the following sub-entities: `BNE_QUERIES`, `BNE_PARAM_LISTS`, `BNE_COMPONENTS`, `BNE_SECURED_OBJECTS`, and `BNE_VIEWERS`.

> **Note:** To ensure that your customizations are handled appropriately in conjunction with the Online Patching feature introduced in Release 12.2, follow the instructions in *Developing and Deploying Customizations in Oracle E-Business Suite Release 12.2*, My Oracle Support Document 1577661.1.

Downloading Integrator Definitions

To download integrator definitions using the Generic Loader, specify the `bneintegrator.lct` configuration file, the name of an LDT file in which to save the downloaded data, and the `BNE_INTEGRATORS` entity. You can choose to download all integrators for an application, a single specific integrator, or two or more specific integrators.

- To download all integrators for an application, specify the application short name for that application in the INTEGRATOR_ASN download parameter. For example:

  ```
  $ FNDLOAD apps 0 Y DOWNLOAD bneintegrator.lct bneintegrators.ldt
  BNE_INTEGRATORS INTEGRATOR_ASN=BNE
  ORACLE Password:
  ```

- To download a single specific integrator, specify the application short name for that application in the INTEGRATOR_ASN download parameter and the internal name of the integrator in the INTEGRATOR_CODE download parameter. For example:

  ```
  $ FNDLOAD apps 0 Y DOWNLOAD bneintegrator.lct bneNewMsgs.ldt
  BNE_INTEGRATORS INTEGRATOR_ASN=BNE
  INTEGRATOR_CODE=FND_NEW_MESSAGES_INTG
  ORACLE Password:
  ```

- To download two or more specific integrators, specify the application short name for that application in the INTEGRATOR_ASN download parameter and a list of the internal names of the integrators in the INTEGRATOR_CODE_LIST download parameter. The integrator internal names in the list should be separated by commas and should not include wildcards. For example:

  ```
  $ FNDLOAD apps 0 Y DOWNLOAD bneintegrator.lct bneNewMsgs.ldt
  BNE_INTEGRATORS INTEGRATOR_ASN=BNE
  INTEGRATOR_CODE_LIST=FND_NEW_MESSAGES_INTG,JOURNALS_115
  ORACLE Password:
  ```

  > **Note:** When you download integrators with the INTEGRATOR_CODE_LIST parameter, the Generic Loader may download any integrators whose internal names are substrings of the specified integrator internal names, in addition to the specified integrators. For example, if you specify INTEGRATOR_CODE_LIST=XYZ_INTEG01,XYZ_INTEG02, the Generic Loader may download integrators named XYZ_INTEG and XYZ_INT as well as the integrators XYZ_INTEG01 and XYZ_INTEG02.

> **Note:** Do not specify both the INTEGRATOR_CODE parameter and the INTEGRATOR_CODE_LIST parameter in the same command. Use only one of these parameters at a time, depending on the number of integrators you want to download.

Uploading Integrator Definitions

To upload integrator definitions using the Generic Loader, specify the bneintegrator.lct configuration file, the name of the LDT file that contains the definition or definitions to upload, and - to upload all entities in the LDT file. For example:

```
$ FNDLOAD apps 0 Y UPLOAD bneintegrator.lct bneintegrators.ldt -
ORACLE Password:
```

Related Topics

Generic Loader*, Oracle E-Business Suite Setup Guide*

Index