

# **TIBCO ActiveMatrix BusinessWorks™ Plug-in for Oracle E-Business Suite User's Guide**

*Software Release 6.1  
December 2015*

## Important Information

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document contains confidential information that is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, Two-Second Advantage, TIBCO ActiveMatrix BusinessWorks, TIBCO ActiveMatrix BusinessWorks Plug-in for Oracle E-Business Suite, TIBCO Business Studio, and TIBCO Enterprise Administrator are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Enterprise Java Beans (EJB), Java Platform Enterprise Edition (Java EE), Java 2 Platform Enterprise Edition (J2EE), and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle Corporation in the U.S. and other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

THIS SOFTWARE MAY BE AVAILABLE ON MULTIPLE OPERATING SYSTEMS. HOWEVER, NOT ALL OPERATING SYSTEM PLATFORMS FOR A SPECIFIC SOFTWARE VERSION ARE RELEASED AT THE SAME TIME. SEE THE README FILE FOR THE AVAILABILITY OF THIS SOFTWARE VERSION ON A SPECIFIC OPERATING SYSTEM PLATFORM.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

Copyright © 2010-2015 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

# Contents

---

<b>TIBCO Documentation and Support Services</b> .....	<b>5</b>
<b>Plug-in Overview</b> .....	<b>6</b>
<b>Oracle E-Business Suite Overview</b> .....	<b>7</b>
<b>Connecting to Oracle Database Server</b> .....	<b>8</b>
<b>Oracle E-Business Suite SQL Scripts</b> .....	<b>11</b>
<b>Getting Started</b> .....	<b>12</b>
Creating a Project .....	12
Creating an Oracle E-Business Suite Connection .....	13
Configuring a Process .....	14
Testing a Process .....	15
Deploying an Application .....	16
TIBCO Business Studio Overview .....	16
<b>Oracle E-Business Suite Connection</b> .....	<b>18</b>
<b>Oracle E-Business Suite Palette</b> .....	<b>21</b>
Oracle API .....	21
Custom API .....	25
Oracle Concurrent Program .....	30
Custom Concurrent Program .....	34
Oracle Business Event .....	38
<b>Working with Sample Projects</b> .....	<b>44</b>
Importing Sample Projects .....	44
Working with the OracleAPI Project .....	45
Running the EGO_ITEM_PUB.bwp Process .....	47
EGO_ITEM_PUB Package Configuration .....	50
Running the BOM_BO_PUB.bwp Process .....	52
BOM_BO_PUB Package Configuration .....	55
Working with the CustomAPI Project .....	56
Running the CustomAPI Project .....	57
ACCOUNT_MGR Package Configuration .....	58
Working with the OracleCP Project .....	58
Running the OracleCP Project .....	59
WICMLP Concurrent Program Configuration .....	62
Working with the CustomCP Project .....	64
Running the CustomCP Project .....	65
INCOIN Concurrent Program Configuration .....	69
Working with the BusinessEvent Project .....	72
Running the BusinessEvent Project .....	72
<b>Managing Logs</b> .....	<b>76</b>

Log Levels .....	76
Setting up Log Levels .....	76
Exporting Logs to a File .....	77
<b>Updating the Oracle JDBC Library or Oracle Advanced Queuing Library .....</b>	<b>78</b>
<b>Backing Up TIB_BW_EBS_LOG Table .....</b>	<b>79</b>
Log Table Structure .....	79
Exporting Data from Oracle E-Business Suite 12.1.x .....	81
Importing Data into Oracle E-Business Suite 12.2.x .....	81
<b>Oracle API and Custom API Comparison .....</b>	<b>83</b>
<b>Oracle Concurrent Program and Custom Concurrent Program Comparison .....</b>	<b>84</b>
SRS Concurrent Program and Non-SRS Concurrent Program .....	84
Sample Non-SRS Concurrent Program: GLBBSU .....	85
Checking the Type and Parameters of a Concurrent Program .....	85
Checking the Type of a Concurrent Program .....	85
Finding the Parameters for a Concurrent Program .....	86
Submitting a Request by Using Oracle E-Business Suite Client .....	87
Submitting a Standard Request Form .....	87
Submitting a Non-Standard Request Form .....	88
Submitting a Request by Using Plug-in Activities .....	91
Using the Oracle Concurrent Program Activity .....	91
Using the Custom Concurrent Program Activity .....	93
<b>Introduction to Oracle Integration Repository .....</b>	<b>96</b>
Searching for a Business Service Interface in Oracle Integration Repository .....	96
<b>Introduction to Oracle Database JPublisher .....</b>	<b>99</b>
Wrapper and Non-Wrapper Data Types .....	99
Oracle JPublisher Usage in the Plug-in .....	100
Installing JPublisher .....	100
Supported Versions .....	101
Using JPublisher .....	102
Granting Privileges to the Running Account .....	102
Running JPublisher .....	102
JPublisher Parameters .....	103
JPublisher Wrapper Details .....	103
<b>Plug-in Error Codes .....</b>	<b>105</b>

# TIBCO Documentation and Support Services

---

Documentation for this and other TIBCO products is available on the TIBCO Documentation site:

<https://docs.tibco.com>

Documentation on the TIBCO Documentation site is updated more frequently than any documentation that might be included with the product. To ensure that you are accessing the latest available help topics, please visit <https://docs.tibco.com>.

## Product-Specific Documentation

Documentation for TIBCO products is not bundled with the software. Instead, it is available on the TIBCO Documentation site. To directly access documentation for this product, double-click the following file:

```
TIBCO_HOME\release_notes\TIB_bwpluginoraclelebs_version_docinfo.html
```

where TIBCO\_HOME is the top-level directory in which TIBCO products are installed. On Windows, the default TIBCO\_HOME is C:\Program Files\tibco. On UNIX systems, the default TIBCO\_HOME is /opt/tibco.

The following documents for this product can be found in the TIBCO Documentation site:

- *TIBCO ActiveMatrix BusinessWorks Plug-in for Oracle E-Business Suite User's Guide* Read this manual for the concepts relating to the plug-in, the applications with which the plug-in interacts, configuration and deployment information, and examples.
- *TIBCO ActiveMatrix BusinessWorks Plug-in for Oracle E-Business Suite Installation* Read this manual for instructions on site preparation and installation.
- *TIBCO ActiveMatrix BusinessWorks Plug-in for Oracle E-Business Suite Release Notes* Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

## How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, contact TIBCO Support:

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit this site:

<http://www.tibco.com/services/support>

- If you already have a valid maintenance or support contract, visit this site:

<https://support.tibco.com>

Entry to this site requires a user name and password. If you do not have a user name, you can request one.

## How to Join TIBCOcommunity

TIBCOcommunity is an online destination for TIBCO customers, partners, and resident experts. It is a place to share and access the collective experience of the TIBCO community. TIBCOcommunity offers forums, blogs, and access to a variety of resources. To register, go to the following web address:

<https://www.tibcommunity.com>

## Plug-in Overview

---

TIBCO ActiveMatrix BusinessWorks™ Plug-in for Oracle E-Business Suite plugs into TIBCO ActiveMatrix BusinessWorks™. You can use this plug-in to configure a connection to Oracle E-Business Suite, and then use activities to integrate with Oracle E-Business Suite.

TIBCO ActiveMatrix BusinessWorks is a leading integration platform that can integrate a wide variety of technologies and systems within enterprise and on cloud. TIBCO ActiveMatrix BusinessWorks includes an Eclipse-based graphical user interface (GUI) provided by TIBCO Business Studio™ for design, testing, and deployment. If you are not familiar with TIBCO ActiveMatrix BusinessWorks, see the TIBCO ActiveMatrix BusinessWorks documentation for more details.

TIBCO ActiveMatrix BusinessWorks Plug-in for Oracle E-Business Suite extends the palette functionality provided by TIBCO ActiveMatrix BusinessWorks. After installing the plug-in, an [Oracle E-Business Suite Connection](#) and an [Oracle E-Business Suite Palette](#) become available in TIBCO Business Studio. You can add the plug-in activities in the palette to the business processes you are designing, and integrate them into the process flow. At run time, the plug-in activities are executed as part of the TIBCO ActiveMatrix BusinessWorks process execution.

TIBCO ActiveMatrix BusinessWorks Plug-in for Oracle E-Business Suite provides the following features:

- Support for Oracle E-Business Suite. (See the readme file for supported versions.)
- Support for Oracle Business Events as outbound Business Objects.
- Support for both PL/SQL API from Oracle Integration Repository and API defined by the customer.
- Support for both Concurrent Programs from Oracle Integration Repository and Concurrent Programs defined by the customer.
- Support for Custom Oracle Business Events. You can register a user-created event in Oracle E-Business Suite, and the registered user-created event becomes searchable in Oracle Integration Repository.



To resolve the character set conversion issues that are caused when mapping Oracle object types (for example, an event type used in a queue table) to Java object types, you can copy the `ora118n.jar` file which contains the classes used for NLS support in Oracle objects and collection types to the `TIBCO_HOME\bw\palettes\oraclebs\version_number\lib` directory. You can get the `Orai118n.jar` file from <http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-10201-088211.html> or the `ORACLE_DB_HOME\jlib` directory.

# Oracle E-Business Suite Overview

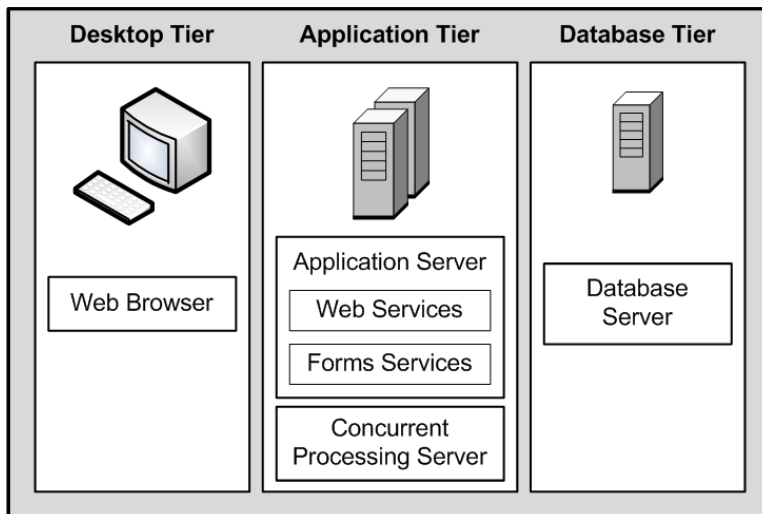
Oracle E-Business Suite (EBS) is a complete set of business applications for managing and automating processes for your enterprise. It is also known as Oracle Enterprise Resource Planning (ERP), Oracle Apps, Oracle Applications, and Oracle Financials on the market.

It is a comprehensive suite of integrated, global business applications that provides the following functions:

- A complete, integrated business intelligence portfolio
- An adaptable global business platform
- A customer-focused applications strategy

## Oracle E-Business Suite Architecture

The Oracle E-Business Suite Architecture is a framework for multi-tiered, distributed computing that supports Oracle E-Business Suite products. In this model, various servers or services are distributed among three levels or tiers.



## Connecting to Oracle Database Server

To configure the basic settings of the plug-in, you must connect the Oracle Database Server by executing two SQL scripts, `create_user.sql` and `common_all.sql`.



You must have an Oracle Database system administrator's account to create a plug-in user and an Oracle E-Business Suite application administrator's account to grant privileges to the plug-in user.

TIBCO ActiveMatrix BusinessWorks Plug-in for Oracle E-Business Suite adds support for Oracle E-Business Suite 12.2.x. Different `create_user.sql` scripts and `common_all.sql` scripts are required if you use the plug-in with different versions of Oracle E-Business Suite. When connecting to Oracle Database Server, select the corresponding scripts according to the version of Oracle E-Business Suite you are using.

- The two scripts for Oracle E-Business Suite 12.2.x are located in the `TIBCO_HOME\bw\palettes\oraclelebs\version_number\config\12.2` directory.
- The two scripts for Oracle E-Business Suite 12.1.x and 12.0.x are located in the `TIBCO_HOME\bw\palettes\oraclelebs\version_number\config\12.1` directory.

See [Oracle E-Business Suite SQL Scripts](#) for more details about the SQL scripts.

### Procedure

1. Log on to Oracle SQL\*Plus:
  - a) On the command line, enter `sqlplus`.
  - b) Enter the Oracle Database system administrator's user name and password.

The following figure shows a sample logon screen:

```

C:\Windows\system32\cmd.exe - sqlplus system/manager@vis_192.168.71.209
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\hogao>sqlplus system/manager@vis_192.168.71.209

SQL*Plus: Release 11.2.0.1.0 Production on Mon Mar 17 16:19:09 2014

Copyright (c) 1982, 2010, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.7.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL>
  
```

2. Create a plug-in user.

The plug-in requires a user that can access various schemas, objects, and data in the Oracle E-Business Suite system. Run the `create_user.sql` file that is located in the `TIBCO_HOME\bw\palettes\oraclelebs\version_number\config\version_number` directory. Enter the user name and password on the command line.



```

C:\Windows\system32\cmd.exe - sqlplus system/manager@vis_192.168.71.209

C:\Users\hogao>sqlplus system/manager@vis_192.168.71.209

SQL*Plus: Release 11.2.0.1.0 Production on Mon Mar 17 16:19:09 2014

Copyright (c) 1982, 2010, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.7.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> @c:\tibco\bw\palettes\oracleebs\6.0\config\create_user.sql
Enter value for plugin_username: plugin
Enter value for plugin_password: plugin

User created.

Grant succeeded.

Grant succeeded.

SQL>

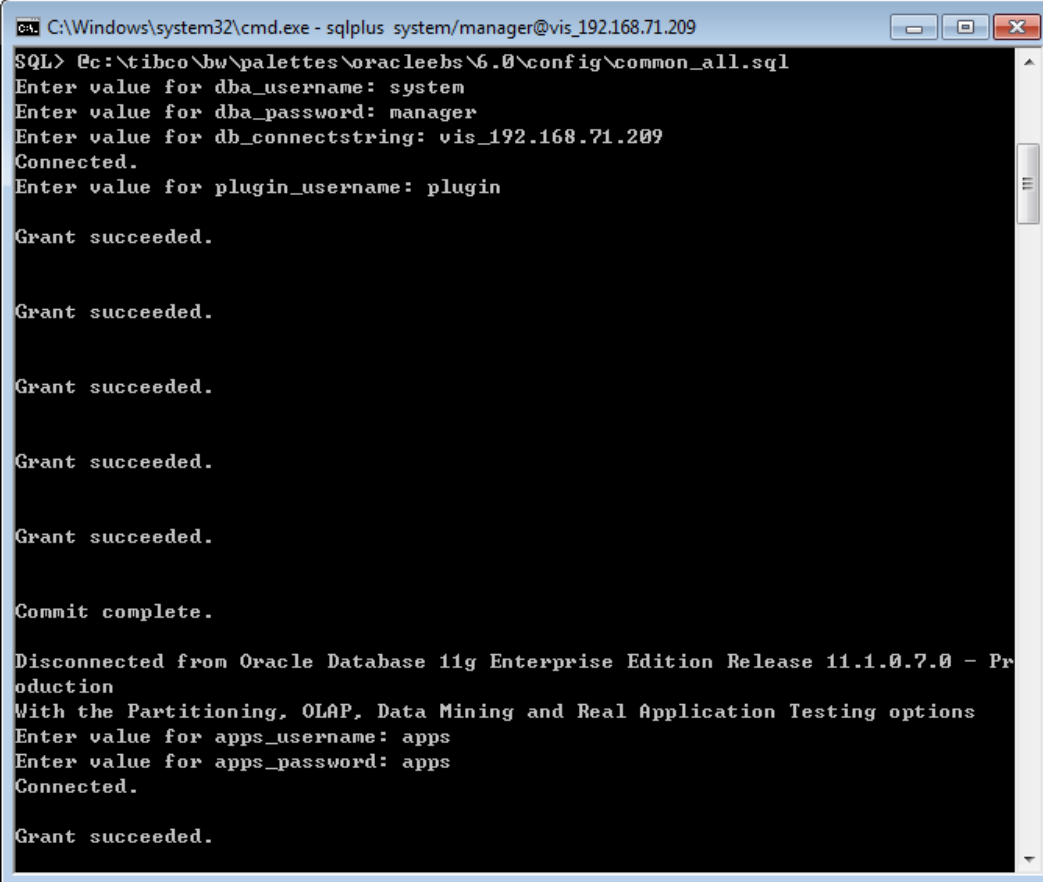
```

### 3. Initialize the plug-in user.

Run the `common_all.sql` file that is located in the `TIBCO_HOME\bw\palettes\oracleebs\version_number\config\version_number` directory to initialize the plug-in user. You are prompted to enter values for the following parameters:

- **dba\_username:** typically system
- **dba\_password:** typically manager
- **db\_connectstring:** net service name
- **plug-in username:** as created in [Step 2](#)
- **plug-in password:** as created in [Step 2](#)
- **apps\_username:** typically apps
- **apps\_password:** typically apps

The following figure shows the script execution screen:



```
C:\Windows\system32\cmd.exe - sqlplus system/manager@vis_192.168.71.209
SQL> @c:\tibco\bw\palettes\oracle\6.0\config\common_all.sql
Enter value for dba_username: system
Enter value for dba_password: manager
Enter value for db_connectstring: vis_192.168.71.209
Connected.
Enter value for plugin_username: plugin

Grant succeeded.

Grant succeeded.

Grant succeeded.

Grant succeeded.

Grant succeeded.

Commit complete.

Disconnected from Oracle Database 11g Enterprise Edition Release 11.1.0.7.0 - Pr
oduction
With the Partitioning, OLAP, Data Mining and Real Application Testing options
Enter value for apps_username: apps
Enter value for apps_password: apps
Connected.

Grant succeeded.
```

# Oracle E-Business Suite SQL Scripts

---

SQL scripts set up the infrastructure in the Oracle E-Business Suite database that is used by some of the features of the plug-in. You can use the scripts to create a plug-in user, grant privileges, revoke privileges, and alter the enabled editions of the plug-in user.

The following SQL scripts can help with your configuration of the basic settings of the plug-in:

- `create_user.sql`

This script creates the database user that the plug-in uses to interact with the database. In this way, you can apply more detailed control over the privileges granted to the plug-in.

For Oracle E-Business Suite 12.2.x, the created plug-in user is only used to issue SQL scripts for Oracle Concurrent Program, Custom Concurrent Program, and Oracle E-Business Event activities at run time.

For Oracle E-Business Suite 12.1.x and 12.0.x, the created plug-in user is used to issue SQL scripts for all plug-in activities at run time.

- `common_all.sql`

This script grants the plug-in user with privileges to access the Oracle Database Dictionary tables, Oracle Integration Repository tables, and Oracle Concurrent Program infrastructure packages and procedures. This script also creates the log table and SQL package used by the plug-in to record the progress and result of every transaction.

- `common_all_undo.sql`

This script reverses all changes made by the `common_all.sql` script. After issuing this script, all privileges granted to the plug-in user are revoked, and the log tables and SQL packages used by the plug-in are deleted.

- `user_upgrade.sql`

This script alters the enable editions of the plug-in user. Issue this script if you upgrade Oracle E-Business Suite 12.1.x to Oracle E-Business Suite 12.2.x.

# Getting Started

---

This tutorial is designed for beginners who want to use TIBCO ActiveMatrix BusinessWorks Plug-in for Oracle E-Business Suite in TIBCO Business Studio.

All operations are performed in TIBCO Business Studio. See [TIBCO Business Studio Overview](#) to get familiar with TIBCO Business Studio.

A basic procedure of using TIBCO ActiveMatrix BusinessWorks Plug-in for Oracle E-Business Suite includes the following steps:

1. [Creating a Project](#)
2. [Creating an Oracle E-Business Suite Connection](#)
3. [Configuring a Process](#)
4. [Testing a Process](#)
5. [Deploying an Application](#)

## Creating a Project

The first task using the plug-in is creating a project. After creating a project, you can add resources and processes.

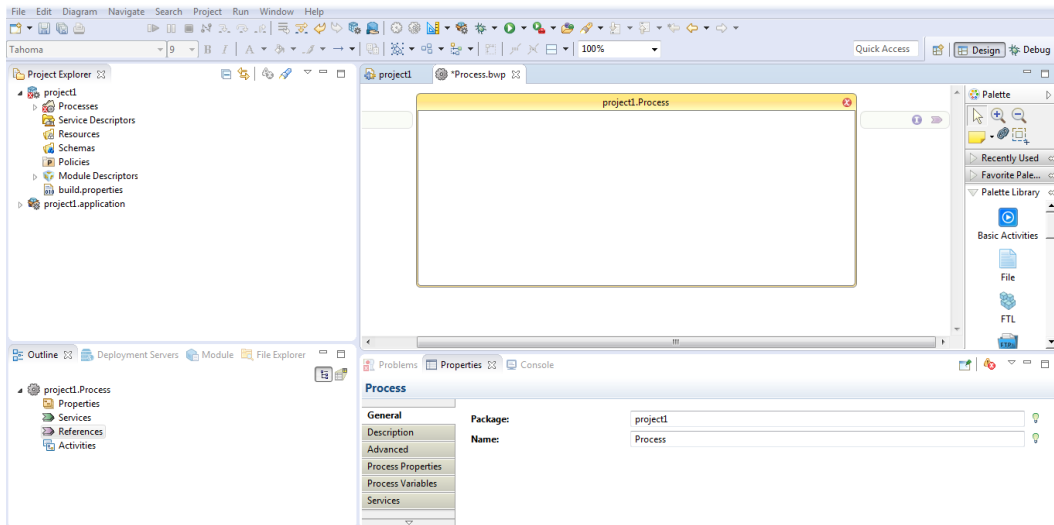
An Eclipse project is an application module configured for TIBCO ActiveMatrix BusinessWorks. An application module is the smallest unit of resources that is named, versioned, and packaged as part of an application.

### Procedure

1. Start TIBCO Business Studio by using one of the following ways:
  - Microsoft Windows: click **Start > All Programs > TIBCO > TIBCO\_HOME > TIBCO Business Studio version\_number > Studio for Designers**.
  - Mac OS and Linux: run the TIBCO Business Studio executable file located in the `TIBCO_HOME/studio/version_number/eclipse` directory.
2. From the menu, click **File > New > BusinessWorks Resources** to open the BusinessWorks Resource wizard.
3. In the "Select a wizard" dialog, click **BusinessWorks Application Module**, and click **Next** to open the New BusinessWorks Application Module wizard.
4. In the Project dialog, configure the project that you want to create:
  - a) In the **Project name** field, enter a project name.
  - b) By default, the created project is located in the workspace current in use. If you do not want to use the default location for the project, clear the **Use default location** check box, and click **Browse** to select a new location.
  - c) Use the default version of the application module, or enter a new version in the **Version** field.
  - d) Keep the **Create empty process** and **Create Application** check boxes selected to automatically create an empty process and an application when creating the project.
  - e) Select the **Use Java configuration** check box if you want to create a Java module.  
A Java module provides Java tooling capabilities.
  - f) Click **Finish** to create the project.

## Result

The project with the specified settings is displayed in the Project Explorer view, as shown in the following figure.



## Creating an Oracle E-Business Suite Connection

After you create a project, you create and configure a connection resource to connect with Oracle E-Business Suite.

### Prerequisites

Before creating the Oracle E-Business Suite shared resource, make sure that an Oracle Database server is connected and a project is created. See [Connecting to Oracle Database Server](#) and [Creating a Project](#) for more details.

### Procedure

1. Expand the created project in the Project Explorer view.
2. Right-click the **Resources** folder, and select **New > OEBSConnection**.
3. In the OEBS Connection wizard, the resource folder, package name, and resource name of the OEBS connection are provided by default. If you do not want to use the default configurations, change them accordingly. Click **Finish**.

- Configure the connection resource in the OEBS Connection editor. See [Oracle E-Business Suite Connection](#) for details about the configuration fields.

- Click **Test Connection** to validate the connection.
- Click **File > Save** to save the project.

## Result

A connection resource is created: `connection_name.oebconnectionResource`. The connection resource is saved in the `Resources\project_name` directory by default or the directory you specified.

## Configuring a Process

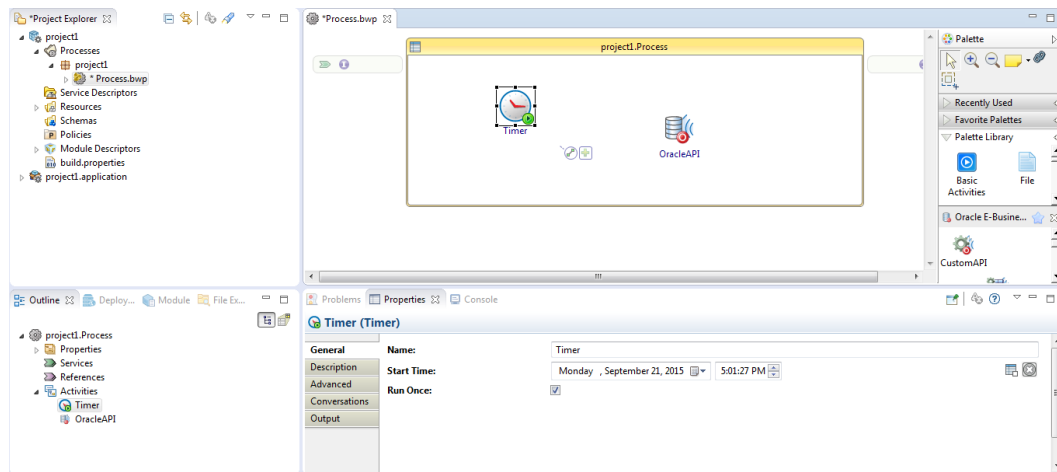
After creating a project, an empty process is created by default. Processes capture and manage the flow of business information in an enterprise between different data sources and destinations. You can configure a process by adding activities.


### Prerequisites

Ensure that you have created an empty process when creating a project. See [Creating a Project](#) for more details.

### Procedure

- In the Project Explorer view, expand the created project, and open the empty process from the Processes folder.
- Select an activity from the Palette view, and drop it in the Process editor.  
For example, select and drag the Timer activity from the General Activities palette and the Oracle API activity from the Oracle E-Business Suite palette.



3. Drag the  icon to create a link between the activities.
4. Configure the added activities as described in [Oracle E-Business Suite Palette](#).



An OEBS connection is required when configuring the OEBS activities. See [Creating an Oracle E-Business Suite Connection](#) for more details about how to create an OEBS connection.

5. Click **File > Save** to save the project.

## Result

The process with configured activities is saved: *process\_name*.bwp. The process is saved in the `Processes\project_name` directory by default or the directory you specified.


## Testing a Process

After configuring a process, you can test the process to check whether the process completes your task.

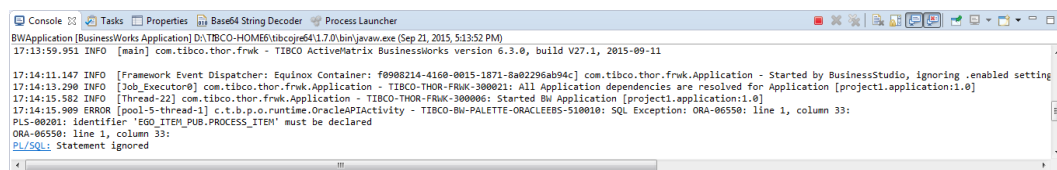
### Prerequisites

Ensure that you have configured the process, as described in [Configuring a Process](#).

### Procedure

1. In the toolbar, click **Run > Debug Configurations**, or click  **Debug > Debug Configurations**.
2. Click **BusinessWorks Application > BWApplication** in the left panel of the opened dialog. By default, all applications in the current workspace are selected in the **Applications** tab. Ensure that only the application you want to debug is selected in the **Applications** tab in the right panel.
3. Click **Debug** to test the process in the selected application.

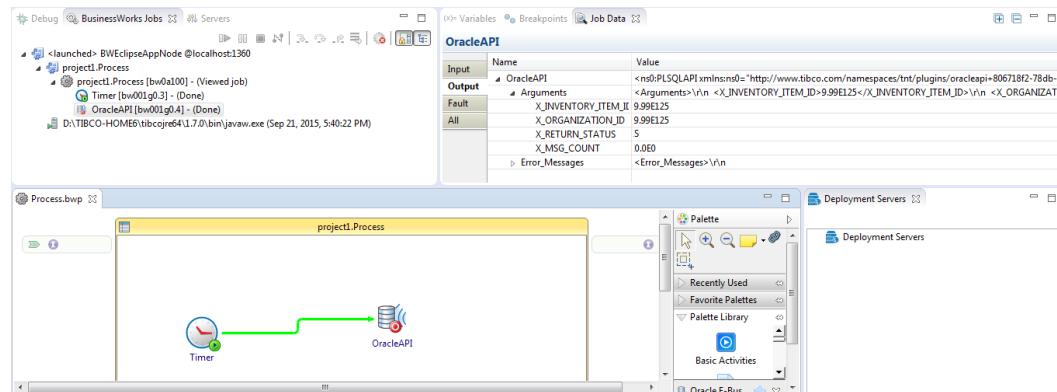
TIBCO Business Studio switches to the Debug perspective. The debug information is displayed in the Console view, as shown in the following figure.



4. In the **Debug** tab in the upper-left corner, expand the running process, and click an activity.

- In the upper-right corner, click the **Job Data** tab, and then click the **Output** tab to check the activity output.

The following figure shows a sample of the activity output:



## Deploying an Application

After testing the process, if the configured process works as expected, you can deploy the application that contains the configured process into a runtime environment, and then use the **bwadmin** utility to manage the deployed application.



Before deploying an application, you must generate an application archive, which is an enterprise archive (EAR) file, in **TIBCO Business Studio**. See *TIBCO ActiveMatrix BusinessWorks Administration* for more details about how to generate application archives.

Deploying an application involves the following tasks:

- Uploading an application archive.
- Deploying an application archive.
- Starting an application.

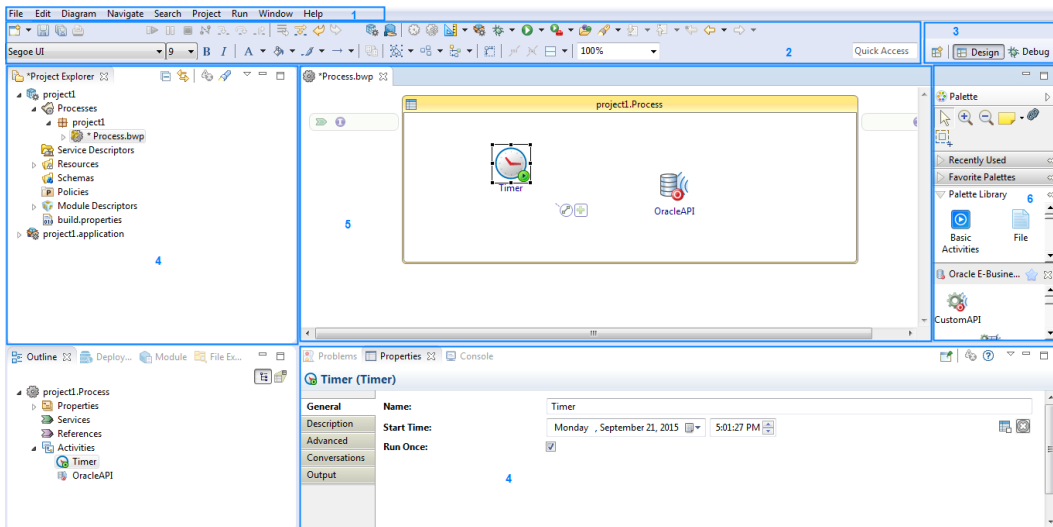
See *TIBCO ActiveMatrix BusinessWorks Administration* for more details about how to deploy an application.

## TIBCO Business Studio Overview








TIBCO Business Studio is an Eclipse-based integration development environment that is used to design, develop, and test ActiveMatrix BusinessWorks applications.

TIBCO Business Studio provides a workbench in which you can create, manage, and navigate resources in your workspace. A *workspace* is the central location on your machine where all data files are stored.





The workbench consists of the following elements:

1. **Menu:** contains menu items, including **File**, **Edit**, **Diagram**, **Navigate**, **Search**, **Project**, **Run**, **Window**, and **Help**.
2. **Toolbar:** contains buttons for frequently used commands, including **New** , **Save** , **Enable/Disable Business Studio capabilities** , **Create a new BusinessWorks Application Module** , **Create a new BusinessWorks Shared Module** , **Debug** , **Run** , and so on.
3. **Perspective:** contains an initial set and layout of views that are required to perform a certain task. TIBCO Business Studio launches the Design perspective by default. You can change the perspective from the menu **Window > Open Perspective > Perspective\_Name**.
4. **View:** displays resources. For example, the Project Explorer view displays the ActiveMatrix BusinessWorks applications, modules, and other resources in your workspace, and the Properties view displays the properties for the selected resource. You can open a view from the menu **Window > Show View > View\_Name**.
5. **Editor:** provides a canvas to configure, edit, or browse a resource. Double-click a resource in a view to open the appropriate editor for the selected resource. For example, double-click an ActiveMatrix BusinessWorks process in the Project Explorer view to open the process in the editor.
6. **Palette:** contains a set of widgets and a palette library. A *palette* groups activities that perform similar tasks, and provides quick access to activities when configuring a process.

## Oracle E-Business Suite Connection

The OEBS Connection resource contains the information that is necessary to establish a connection pool at run time to the Oracle E-Business Suite server, and is used by the activities to obtain connections to the server.

### General Panel

In the **General** panel, you can specify the package that stores the OEBS Connection shared resource and the shared resource name.

The **General** panel contains the following fields:


Field	Module Property?	Description
<b>Package</b>	No	The name of the package where the new shared resource is added.
<b>Name</b>	No	The name of the Oracle E-Business Suite connection. The default is OEBSConnectionResource.
<b>Description</b>	No	A short description of the resource.

### OEBS Connection Configuration Panel

In the **OEBS Connection Configuration** panel, you can provide the information of the OEBS server to which the plug-in connects.

The **OEBS Connection Configuration** panel contains the following fields:

Field	Module Property?	Description
<b>Database URL</b>	Yes	The standard Oracle JDBC connection string. For example, <code>jdbc:oracle:thin:@192.168.71.209:1521:vis</code>

Field	Module Property?	Description
<b>APPS User Name</b>	Yes	<p>The user name of an Oracle E-Business Suite application administrator used to access Oracle E-Business Suite.</p> <p>For Oracle E-Business Suite 12.2.x, the apps user is used to connect to the database server for all plug-in activities at design time, get the version number of Oracle E-Business Suite for the Oracle API, Custom API, and Oracle Business Event activities at run time, and run the Oracle API and Custom API activities at run time.</p> <p>For Oracle E-Business Suite 12.1.x and 12.0.x, the apps user is used to connect to the database server for all plug-in activities at design time, and get the version number of Oracle E-Business Suite for the Oracle API, Custom API, and Oracle Business Event activities at run time; but the plug-in user is used to run all plug-in activities at run time.</p>
<b>APPS User Password</b>	Yes	<p>The password of an Oracle E-Business Suite application administrator used to access Oracle E-Business Suite.</p>
<b>Plugin User Name</b>	Yes	<p>The user name used to run the plug-in activities at run time.</p> <p>For Oracle E-Business Suite 12.2.x, the plug-in user is used to run the Oracle Concurrent Program, Custom Concurrent Program, and Oracle Business Event activities at run time.</p> <p>For Oracle E-Business Suite 12.1.x and 12.0.x, the plug-in user is used to run all plug-in activities at run time.</p>
<b>Plugin User Password</b>	Yes	<p>The password of the plug-in user used to run the plug-in activities at run time.</p>
<b>Maximum Number of Reconnect Attempts</b>	Yes	<p>The maximum number of reconnection attempts to make when the connection is lost.</p> <p>The default value is 3.</p> <div data-bbox="858 1577 1497 1709" style="border-left: 1px solid black; padding-left: 10px;">  If you set the value of this field to a negative integer, the plug-in attempts to reconnect to Oracle E-Business Suite indefinitely when a connection fails. </div>
<b>Interval between Reconnect Attempts (ms)</b>	Yes	<p>The time interval in milliseconds to elapse between reconnection attempts.</p> <p>The default value is 5000, and the minimum value is 0.</p>

Field	Module Property?	Description
<b>Maximum Connections</b>	Yes	<p>The maximum number of connections between the plug-in and Oracle E-Business Suite that are maintained in the plug-in connection pool.</p> <p>The importance of this number becomes clear at run time. When a project has more than one process simultaneously triggered at run time, the processes might request more connections than are allowed by the maximum number. In this case, some processes are suspended until free connections are released.</p> <p>The default value is 3, and the minimum value is 1.</p>
<b>Login Timeout (s)</b>	Yes	<p>The timeout interval in seconds required to log in.</p> <p>The default value is 30, and the minimum value is 0.</p>
<b>Test Connection</b>	No	<p>Used to test whether the connection to Oracle E-Business Suite can be established.</p> <p>This operation tests whether the JDBC libraries are correctly referenced and whether the provided credentials are correct.</p>

# Oracle E-Business Suite Palette

A palette groups the activities that connect the same external applications together. An Oracle E-Business Suite palette is added after installing TIBCO ActiveMatrix BusinessWorks Plug-in for Oracle E-Business Suite. The Oracle E-Business Suite palette includes several activities to be integrated with Oracle E-Business Suite.

The Oracle E-Business Suite palette contains the following activities which encapsulate routing rules for communication messages and custom configuration information:

- [Oracle API](#)
- [Custom API](#)
- [Oracle Concurrent Program](#)
- [Custom Concurrent Program](#)
- [Oracle Business Event](#)

## Oracle API

You can use the Oracle API activity to call the PL/SQL APIs that are listed in Oracle Integration Repository. This activity can inspect the Integration Repository for parameter input and output information, and present this information in a user interface for configuration.



A comparison between Oracle API and Custom API is provided in [Oracle API and Custom API Comparison](#).

The Oracle API activity provides the following functions:

- At design time
  - Loads the PL/SQL procedure information from Oracle Integration Repository.
  - Saves this information to a file.
- At run time
  - Loads the PL/SQL procedure information from the file.
  - Generates a calling SQL statement.
  - Calls the PL/SQL procedure.



Oracle Integration Repository is shipped with Oracle E-Business Suite 12.0.x, 12.1.x, and 12.2.x; therefore, Oracle API can be used in these versions. See [Introduction to Oracle Integration Repository](#) for more details.

### General



In the **General** tab, you can establish a connection to an Oracle E-Business Suite server, specify the API package name and procedure name, and set the timeout time for calling the procedure.

When searching for the API package name, API procedure name, wrapper package name, and wrapper procedure name, it is good practice to enter the name prefixes or as much as you know about the names in the name fields before clicking the search buttons to reduce the search time.



The **API Procedure Name** label in the **General** tab refers to the API procedure or function. It is labeled as **API Procedure Name** because of the limited space available.

The **General** tab contains the following fields:

Field	Module Property?	Description
<b>Name</b>	No	The name of the resource. The default is OracleAPI.
<b>Shared Connection</b>	Yes	<p>A shared connection resource containing the Oracle E-Business Suite connection information.</p> <p>To select a shared connection resource, click the <b>Choose/Create Default Resource</b>  icon. All usable connections are listed in the opened dialog. Select one to apply to your current activity. If no matching connection resource is found, click <b>Create Shared Resource</b> to create one. See <a href="#">Oracle E-Business Suite Connection</a> for more details.</p> <div style="border-left: 1px solid black; padding-left: 10px; margin-left: 20px;"> <p> For Oracle E-Business Suite 12.2.x, the apps user is used to run the Oracle API activities at run time; therefore, before running the Oracle API activity, ensure that the apps user name that you specified in the Shared Connection is correct. For Oracle E-Business Suite 12.1.x and 12.0.x, the plug-in user is used to run the Oracle API activities at run time; therefore, before running the Oracle API activity, ensure that the plug-in user name that you specified in the Shared Connection is correct.</p> </div>
<b>API Package Name</b>	No	<p>A business service interface that is exposed by Oracle E-Business Suite.</p> <p>Click <b>Search Package</b> or the text field. Then, enter a part of the package name in the opened dialog or leave the text field empty, and click <b>Search</b>. Select a package in the opened dialog, and click <b>OK</b>.</p>
<b>API Procedure Name</b>	No	<p>A procedure from the API Package.</p> <p>Click <b>Search Procedure</b> or the text field. Then, enter a part of the procedure name in the opened dialog or leave the text field empty, and click <b>Search</b>. Select a procedure in the opened dialog, and click <b>OK</b>.</p>
<b>Wrapper Package Name</b>	No	<p>The name given to a package of wrapper procedures or functions. This field is displayed only when a wrapper package is required.</p> <p>Click <b>Search Package</b> or the text field. Then, enter a part of the package name in the opened dialog or leave the text field empty, and click <b>Search</b>. Select a package in the opened dialog, and click <b>OK</b>.</p> <p>Some business objects in the Oracle database cannot be accessed through the Oracle JDBC driver directly. In this case, you must create the wrapper package first by using Oracle JPublisher. See <a href="#">Introduction to Oracle Database JPublisher</a> for details about how to create a wrapper package by using JPublisher.</p>

Field	Module Property?	Description
<b>Wrapper Procedure Name</b>	No	<p>The name given to a wrapper procedure or function that is used to delegate an API procedure or function.</p> <p>This field is displayed only when a wrapper procedure or function is required.</p> <p>Click <b>Search Procedure</b> or the text field. Then, enter a part of the procedure name in the opened dialog or leave the text field empty, and click <b>Search</b>. Select a procedure in the opened dialog, and click <b>OK</b>.</p>
<b>Call Procedure Timeout(s)</b>	Yes	<p>The maximum waiting time for calling the procedure.</p> <p>If the procedure call exceeds this limit, the attempt is stopped.</p> <p>The default is 60. Entering 0 results in an unlimited waiting time.</p>
<b>Generate SQL Script</b>	No	<p>After configuring the Oracle API activity, click <b>Generate SQL Script</b>. The scripts to be generated are displayed in the opened dialog. Click <b>Generate</b>, and the following two SQL scripts are created in the <code>TIBCO_HOME\bw\palettes\oracle\eebs\version_number\sql</code> directory:</p> <ul style="list-style-type: none"> <li>• <code>ProcessName_ActivityName_APIPackageName.APIProcedureName.sql</code></li> <li>• <code>ProcessName_ActivityName_APIPackageName.APIProcedureName_undo.sql</code></li> </ul> <p>Run the <code>ProcessName_ActivityName_APIPackageName.APIProcedureName.sql</code> script in the connected Oracle database before starting the process.</p> <p>For Oracle E-Business Suite 12.2.x, the plug-in uses the apps user to run the Oracle API activity at run time; therefore, you do not have to generate and run the SQL scripts.</p> <p>For Oracle E-Business Suite 12.1.x and 12.0.x, the plug-in uses the plug-in user to run the Oracle API activity at run time; therefore, you must generate and run the <code>ProcessName_ActivityName_APIPackageName.APIProcedureName.sql</code> script to grant privileges to the plug-in user so that the plug-in user can access the objects being configured. The plug-in uses the apps user entered in the Oracle E-Business Suite Connection resource to get the information to generate this script. After this script is executed, the plug-in user can access the tables and procedures configured for the activity.</p> <p>The script with the <code>_undo</code> suffix reverses the granted privileges and undoes the changes to the database. With this operation, you can cleanse the privileges for objects that are no longer used by obsolete or unwanted activities.</p>

## Description

In the **Description** tab, you can enter a short description for the Oracle API activity.

## Prerequisite

The **Prerequisite** tab is used to call some Set Client Info and Initialization procedures before calling the main procedure selected in the **General** tab. You can add new procedure calls and their corresponding package information in the **Prerequisite** tab. When the procedure must be wrapped, you must enter a wrapper package, and select a procedure for the wrapper package.



If the procedure name for the wrapper package contains a single dollar sign (\$), the \$ in the procedure name is converted to a hyphen (-) in the **Input** and **Output** tabs.

## PLSQL API Details

In the **PLSQL API Details** tab, you can view the details of the API procedure and wrapper procedure (if wrapper procedure exists), which are configured in the **General** tab. The details are listed in the Parameter Name, Parameter Type, and Parameter Direction columns.

- The following figure shows the PLSQL API details without a wrapper package:

Parameter Name	Parameter Type	Parameter Direction
P_API_VERSION	NUMBER	IN
P_INIT_MSG_LIST	VARCHAR2	IN
P_COMMIT	VARCHAR2	IN
P_TRANSACTION_TYPE	VARCHAR2	IN
P_LANGUAGE_CODE	VARCHAR2	IN
P_TEMPLATE_ID	NUMBER	IN
P_TEMPLATE_NAME	VARCHAR2	IN
P_INVENTORY_ITEM_ID	NUMBER	IN

- The following figure shows the PLSQL API details with a wrapper package:

Parameter Name	Parameter Type	Parameter Direction
▲ BOM_BO_PUB.PROCESS_BOM		
P_BO_IDENTIFIER	VARCHAR2	IN
P_API_VERSION_NUMBER	NUMBER	IN
P_INIT_MSG_LIST	PL/SQL BOOLEAN	IN
P_BOM_HEADER_REC	BOM_BO_PUB.BOM_HEAD_REC_TYPE	IN
P_BOM_REVISION_TBL	BOM_BO_PUB.BOM_REVISION_TBL_TYPE	IN
P_BOM_COMPONENT_TBL	BOM_BO_PUB.BOM_COMPS_TBL_TYPE	IN
P_BOM_REF_DESIGNATOR_TBL	BOM_BO_PUB.BOM_REF_DESIGNATOR_TBL_TYPE	IN
▲ TIB_BOM_BO_PUB.BOM_BO_PUBSPROCESS_BOM		
P_BO_IDENTIFIER	VARCHAR2	IN
P_API_VERSION_NUMBER	NUMBER	IN
P_INIT_MSG_LIST	NUMBER	IN
P_BOM_HEADER_REC	BOM_BO_PUB_BOM_HEAD_REC_TYPE	IN
P_BOM_REVISION_TBL	BOM_BO_PUB_BOM_REVISION_TBL_T	IN
P_BOM_COMPONENT_TBL	BOM_BO_PUB_BOM_COMPS_TBL_TYPE	IN
P_BOM_REF_DESIGNATOR_TBL	BOM_BO_PUB_BOM_REF_DESIGNATOR	IN

If the API procedure contains parameters of PLSQL data types, such as PLSQL record type, PLSQL table type which is defined in the PLSQL package, and BOOLEAN type, the plug-in requires the related wrapper package and procedure generated from Oracle JPublisher. You must execute the wrapper SQL generated by JPublisher under the apps user.

The Parameter Direction column shows whether the parameter is an input, output, or both for the API procedure.

The **Return** parameter in the **PLSQL API Details** tab specifies that `GL_JOURNAL_IMPORT_PKG.GET_LAST_SQL` is an Oracle database function, and the return value can be used as an output.



## Input

In the **Input** tab, you can view the input data for the Oracle API activity. All parameters of the API procedure or wrapper procedure with the **Parameter Direction** IN or IN/OUT in the **PLSQL API Details** tab are listed in the OracleAPI-input column.



The parameters listed in the OracleAPI-input column can be configured manually or automatically by defining a module property. For how to define module properties, see "Using Process and Module Properties" in *TIBCO ActiveMatrix BusinessWorks Samples*.

## Output

In the **Output** tab, the **Arguments** node lists all parameters of the API procedure or wrapper procedure with the **Parameter Direction** OUT or IN/OUT in the **PLSQL API Details** tab, and the **Error\_Messages** node lists the FND (Oracle Application Object Library) error messages when the API encounters errors in the Oracle E-Business Suite system.

## Fault

In the **Fault** tab, you can find the error messages of the Oracle API activity.

See [Plug-in Error Codes](#) for more information about error messages and corrective actions to take.

The **Fault** tab lists the following error messages:

Fault	Thrown When ...
OracleEBSPaletteAQConnectionException	The AQ Connection does not initialize.
OracleEBSPaletteConnectionNotFoundException	The connection cannot be created.
OracleEBSPaletteSQLException	An SQL Exception occurs.
OracleEBSPaletteTimeoutException	Calling the procedure for PLSQL API.

## Custom API

You can use the Custom API activity to call packages that are created by a customer to be integrated with Oracle E-Business Suite not registered with Oracle Integration Repository.

For the Custom API activity, the metadata that defines the package is not contained in Oracle Integration Repository. The plug-in instead uses Oracle Database Dictionary to get detailed information about this API. If you create a Custom API and register it into the Integration Repository, it becomes a regular Oracle E-Business Suite API and is treated as such.



A comparison between Oracle API and Custom API is provided in [Oracle API and Custom API Comparison](#).

The Custom API activity provides the following functions:

- At design time
  - Loads the PL/SQL procedure information from Oracle Database Dictionary.
  - Saves this information to a file.
- At run time
  - Loads the PL/SQL procedure information from the file.

- Generates a calling SQL statement.
- Calls the PL/SQL procedure.





Custom API can be used in Oracle E-Business Suite 12.0.x, 12.1.x, and 12.2.x.

## General

In the **General** tab, you can establish a connection to an Oracle E-Business Suite server, specify the API package name and procedure name, and set the timeout time for calling a procedure.

The **General** tab contains the following fields:

Field	Module Property?	Description
<b>Name</b>	No	The name of the resource. The default is <code>CustomAPI</code> .
<b>Shared Connection</b>	Yes	A shared configuration resource containing the Oracle E-Business Suite connection information. To select a shared connection resource, click the <b>Choose/Create Default Resource</b>  icon. All usable connections are listed in the opened dialog. Select one to apply to your current activity. If no matching connection resource is found, click <b>Create Shared Resource</b> to create one. See <a href="#">Oracle E-Business Suite Connection</a> for more details.   For Oracle E-Business Suite 12.2.x, the apps user is used to run the Custom API activities at run time; therefore, before running the Custom API activity, ensure that the apps user name that you specified in the Shared Connection is correct. For Oracle E-Business Suite 12.1.x and 12.0.x, the plug-in user is used to run the Custom API activities at run time; therefore, before running the Custom API activity, ensure that the plug-in user name that you specified in the Shared Connection is correct.
<b>API Package Name</b>	No	A business service interface that is exposed by Oracle E-Business Suite. Click <b>Search Package</b> or the text field. Then, enter a part of the package name in the opened dialog or leave the text field empty, and click <b>Search</b> . Select a package in the opened dialog, and click <b>OK</b> .
<b>API Procedure Name</b>	No	A procedure from the API Package. Click <b>Search Procedure</b> or the text field. Then, enter a part of the procedure name in the opened dialog or leave the text field empty, and click <b>Search</b> . Select a procedure in the opened dialog, and click <b>OK</b> .

Field	Module Property?	Description
<b>Wrapper Package Name</b>	No	<p>The name given to a package of wrapper procedures or functions. This field is displayed only when a wrapper package is required.</p> <p>Click <b>Search Package</b> or the text field. Then, enter a part of the package name in the opened dialog or leave the text field empty, and click <b>Search</b>. Select a package in the opened dialog, and click <b>OK</b>.</p> <p>Some business objects in the Oracle database cannot be accessed through the Oracle JDBC driver directly. In this case, you must create the wrapper package first by using Oracle JPublisher. See <a href="#">Introduction to Oracle Database JPublisher</a> for details about how to create a wrapper package by using JPublisher.</p>
<b>Wrapper Procedure Name</b>	No	<p>The name given to a wrapper procedure or function that is used to delegate an API procedure or function.</p> <p>This field is displayed only when a wrapper procedure or function is required.</p> <p>Click <b>Search Procedure</b> or the text field. Then, enter a part of the procedure name in the opened dialog or leave the text field empty, and click <b>Search</b>. Select a procedure in the opened dialog, and click <b>OK</b>.</p>
<b>Call Procedure Timeout(s)</b>	Yes	<p>The maximum waiting time for calling the procedure.</p> <p>If the procedure call exceeds this limit, the attempt is stopped.</p> <p>The default is 60. Entering 0 results in an unlimited waiting time.</p>

Field	Module Property?	Description
<b>Generate SQL Script</b>	No	<p>After configuring the Custom API activity, click <b>Generate SQL Script</b>. The scripts to be generated are displayed in the opened dialog. Click <b>Generate</b>, and the following two SQL scripts are created in the <code>TIBCO_HOME\bw\palettes\oracleebs\version_number\sql</code> directory:</p> <ul style="list-style-type: none"> <li>• <code>ProcessName_ActivityName_APIPackageName.APIProcedureName.sql</code></li> <li>• <code>ProcessName_ActivityName_APIPackageName.APIProcedureName_undo.sql</code></li> </ul> <p>Run the <code>ProcessName_ActivityName_APIPackageName.APIProcedureName.sql</code> script in the connected Oracle database before starting the process.</p> <p>For Oracle E-Business Suite 12.2.x, the plug-in uses the apps user to run the Custom API activity at run time; therefore, you do not have to generate and run the SQL scripts.</p> <p>For Oracle E-Business Suite 12.1.x and 12.0.x, the plug-in uses the plug-in user to run the Custom API activity at run time; therefore, you must generate and run the <code>ProcessName_ActivityName_APIPackageName.APIProcedureName.sql</code> script to grant privileges to the plug-in user so that the plug-in user can access the objects being configured. The plug-in uses the apps user entered in the Oracle E-Business Suite Connection resource to get the information to generate this script. After this script is executed, the plug-in user can access the tables and procedures configured for the activity.</p> <p>The script with the <code>_undo</code> suffix reverses the granted privileges and undoes the changes to the database. With this operation, you can cleanse the privileges for objects that are no longer used by obsolete or unwanted activities.</p>

## Description

In the **Description** tab, you can enter a short description for the Custom API activity.

## Prerequisite

The **Prerequisite** tab is used to call some Set Client Info and Initialization procedures before calling the main procedure selected in the **General** tab. You can add new procedure calls and their corresponding package information in the **Prerequisite** tab. When the procedure must be wrapped, you must enter a wrapper package, and select a procedure for the wrapper package.

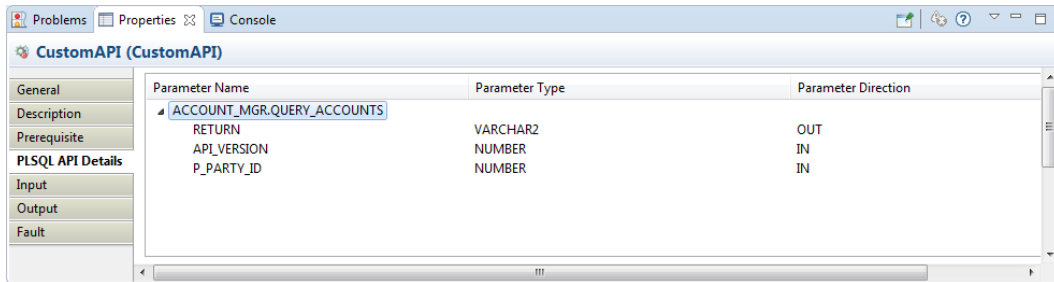


If the procedure name for the wrapper package contains a single dollar sign (\$), the \$ in the procedure name is converted to a hyphen (-) in the **Input** and **Output** tabs.

## PLSQL API Details

In the **PLSQL API Details** tab, you can view the details of the API procedure and wrapper procedure (if wrapper procedure exists), which are configured in the **General** tab. The details are listed in the Parameter Name, Parameter Type, and Parameter Direction columns.

The following figure shows the **PLSQL API Details** tab:



If the API procedure contains parameters of PLSQL data types, such as PLSQL record type, PLSQL table type which is defined in the PLSQL package, and BOOLEAN type, the plug-in requires the related wrapper package and procedure generated from Oracle JPublisher. You must execute the wrapper SQL generated by JPublisher under the apps user.

The Parameter Direction column shows whether the parameter is an input, output, or both for the API procedure.

### Input

In the **Input** tab, you can view the input data for the Custom API activity. All parameters of the API procedure or wrapper procedure with the **Parameter Direction** IN or IN/OUT in the **PLSQL API Details** tab are listed in the CustomAPI-input column.



The parameters listed in the CustomAPI-input column can be configured manually or automatically by defining a module property. For how to define module properties, see "Using Process and Module Properties" in *TIBCO ActiveMatrix BusinessWorks Samples*.

### Output

In the **Output** tab, the **Arguments** node lists all parameters of the API procedure or wrapper procedure with the **Parameter Direction** OUT or IN/OUT in the **PLSQL API Details** tab, and the **Error\_Messages** node lists the FND (Oracle Application Object Library) error messages when the API encounters errors in the Oracle E-Business Suite system.

### Fault

In the **Fault** tab, you can find the error messages of the Custom API activity.

See [Plug-in Error Codes](#) for more information about error messages and corrective actions to take.

The **Fault** tab lists the following error messages:

Fault	Thrown When ...
OracleEBSPaletteAQConnectionException	The AQ Connection does not initialize.
OracleEBSPaletteConnectionNotFoundException	The connection cannot be created.
OracleEBSPaletteSQLException	An SQL Exception occurs.
OracleEBSPaletteTimeoutException	Calling the procedure for PLSQL API.

## Oracle Concurrent Program

You can use the Oracle Concurrent Program activity to populate Oracle interface tables, execute Concurrent Programs, and retrieve status information about the execution.

You can use the Oracle Concurrent Program activity to inspect the Integration Repository for the schema of the interface tables of the selected Concurrent Programs as well as the parameters required to execute the Concurrent Program, and to present this information in a user interface for configuration.



A comparison between Oracle Concurrent Program and Custom Concurrent Program is provided in [Oracle Concurrent Program and Custom Concurrent Program Comparison](#).

The Oracle Concurrent Program activity provides the following functions:

- At design time
  - Loads Concurrent Program parameters from the FND table.
  - Loads Inbound Interface Tables or Views from Oracle Integration Repository.
  - Saves the information to a file.
- At run time
  - Loads Concurrent Program parameters and Interface Tables or Views from the file.
  - Uses JDBC to insert data into Interface Tables or Views.
  - Uses JDBC to submit a Concurrent Program request.





Oracle Integration Repository is shipped with Oracle E-Business Suite 12.0.x, 12.1.x, and 12.2.x; therefore, Oracle Concurrent Program can be used in these versions. See [Introduction to Oracle Integration Repository](#) for more details.


### General

In the **General** tab, you can establish a connection to an Oracle E-Business Suite server, specify the language, concurrent program name, responsibility name, and user name, and set the check interval and total amount of time for checking the result of executing the Concurrent Program.

The **General** tab contains the following fields:

Field	Module Property?	Description
<b>Name</b>	No	The name of the resource. The default is OracleConcurrentProgram.

Field	Module Property?	Description
<b>Shared Connection</b>	Yes	<p>A shared configuration resource containing the Oracle E-Business Suite connection information.</p> <p>To select a shared connection resource, click the <b>Choose/Create Default Resource</b>  icon. All usable connections are listed in the opened dialog. Select one to apply to your current activity. If no matching connection resource is found, click <b>Create Shared Resource</b> to create one. See <a href="#">Oracle E-Business Suite Connection</a> for more details.</p> <div style="border-left: 1px solid black; padding-left: 10px; margin-top: 10px;">  Before running the Oracle Concurrent Program activity, ensure that the plug-in user name that you specified in the Shared Connection is correct. </div>
<b>Language</b>	No	<p>The language in which the Oracle E-Business Suite user interface is shown.</p> <p>Click <b>Search Language</b> or the text field. Then, enter a part of the language name in the opened dialog or leave the text field empty, and click <b>Search</b>. All languages that are installed in Oracle E-Business Suite and match the search criteria are listed in the opened dialog. Select a language in the opened dialog, and click <b>OK</b>. The contents of the concurrent program name, responsibility name, and output change accordingly.</p>
<b>Concurrent Program Name</b>	No	<p>Programs that can be called by a concurrent manager.</p> <p>Click <b>Search Concurrent Program Name</b> or the text field. Then, enter a part of the concurrent program name in the opened dialog or leave the text field empty, and click <b>Search</b>. Select a Concurrent Program from the Concurrent Program List dialog, and click <b>OK</b>.</p>
<b>Responsibility Name</b>	No	<p>A responsibility is a level of authority in Oracle E-Business Suite. You can only access the functions and data specified by the selected responsibility.</p> <p>Click <b>Search Responsibility Name</b> or the text field. Then, enter a part of the responsibility name in the opened dialog or leave the text field empty, and click <b>Search</b>. Select a Concurrent Program from the Concurrent Program List dialog, and click <b>OK</b>.</p>
<b>User Name</b>	No	<p>The name of the user whose information is stored in Oracle E-Business Suite.</p> <p>Click <b>Search User Name</b> or the text field. Then, enter a part of the user name in the opened dialog or leave the text field empty, and click <b>Search</b>. Select a user name from the User Name List dialog, and click <b>OK</b>.</p>
<b>Concurrent Request</b>	No	<p>If this check box is selected, the plug-in inserts records into interface tables, and submits the concurrent request. Otherwise, the plug-in only inserts records into interface tables.</p>

Field	Module Property?	Description
<b>Wait for Response</b>	No	Select this check box if you want the Concurrent Program to retrieve <b>REQUEST_ID</b> , <b>PHASE</b> , <b>STATUS</b> , and <b>MESSAGE</b> information, which are shown in the <b>Output</b> tab. If this check box is cleared, the response time becomes shorter, but only <b>REQUEST_ID</b> is retrieved.
<b>Check Interval (s)</b>	Yes	<p>The time interval for checking the result of executing the Concurrent Program.</p> <p>The default is 15.</p> <div style="border-left: 1px solid black; padding-left: 10px; margin-left: 20px;">  <p>When running the Oracle Concurrent Program activity, if the total waiting time that you set for checking the result of the Concurrent Program activity is shorter than the check interval, the plug-in still checks the result once.</p> </div>
<b>Wait Time (s)</b>	Yes	<p>The total amount of time allowed for checking the result of executing the Concurrent Program.</p> <p>The time set in this field should be a multiple of the time set in the <b>Check Interval (s)</b> field. The default is 0, which means the plug-in waits until the Concurrent Program execution is completed.</p>
<b>Generate SQL Script</b>	No	<p>After configuring the Oracle Concurrent Program activity, click <b>Generate SQL Script</b>. The scripts to be generated are displayed in the opened dialog. Click <b>Generate</b>, and the following two SQL scripts are created in the <code>TIBCO_HOME\bw\palettes\oracle\lebs\version_number\sql</code> directory:</p> <ul style="list-style-type: none"> <li>• <code>ProcessName_ActivityName_ConcurrentProgramName.sql</code></li> <li>• <code>ProcessName_ActivityName_ConcurrentProgramName_undo.sql</code></li> </ul> <p>Run the <code>ProcessName_ActivityName_ConcurrentProgramName.sql</code> script in the connected Oracle database before starting the process.</p> <p>The <code>ProcessName_ActivityName_ConcurrentProgramName.sql</code> script is used to grant privileges to the plug-in user so that the plug-in user can access the objects being configured. The plug-in uses the apps user entered in the Oracle E-Business Suite Connection resource to get the information to generate this script. After this script is executed, the plug-in user can access the tables and procedures configured for the activity.</p> <p>The script with the <code>_undo</code> suffix reverses the granted privileges and undoes the changes to the database. With this operation, you can cleanse the privileges for objects that are no longer used by obsolete or unwanted activities.</p>





If the execution of the Concurrent Program takes a significant amount of time, you can clear the **Wait for Response** check box, and just get **REQUEST\_ID** in the **Output** tab. Or you select the **Wait for Response** check box, and set the **Wait Time (s)** field to a limited time; in this case, the Concurrent Program returns the output data no matter whether the execution of the Concurrent Program is completed or not.

### Description

In the **Description** tab, you can enter a short description for the Oracle Concurrent Program activity.

### Concurrent Program Details

In the **Concurrent Program Details** tab, you can view the procedures in the plug-in user schema and the related interface tables or views (if they exist).

### Input

In the **Input** tab, you can view the input data for the Oracle Concurrent Program activity. The parameters in the OracleConcurrentProgram-input column correspond to those shown in the **Concurrent Program Details** tab, with the following exceptions: **application**, **program**, **description**, **start\_time**, and **sub\_request**.



Select the **Concurrent Request** check box in the **General** tab to show **FND\_REQUEST.SUBMIT\_REQUEST**.

The **ResponsibilityName** and **UserName** are dynamic parameters. They are configured in the **General** tab. You can also change the two parameters in the **Input** tab during run time.



The parameter settings in the **Input** tab takes precedence over those in the **General** tab.

### Output

In the **Output** tab, you can find the output information of the Oracle Concurrent Program activity. If the **Wait for Response** check box is selected in the **General** tab, the **REQUEST\_ID**, **PHASE**, **STATUS**, and **MESSAGE** information are all displayed in the **Output** tab. Otherwise, only **REQUEST\_ID** is retrieved.



If you clear the **Concurrent Request** check box in the **General** tab, the output is empty.

### Fault

In the **Fault** tab, you can find the error messages of the Oracle Concurrent Program activity.

See [Plug-in Error Codes](#) for more information about error messages and corrective actions to take.

The **Fault** tab lists the following error messages:

Fault	Thrown When ...
OracleEBSPaletteAQConnectionException	The AQ Connection does not initialize. A JMSEException happens during an incoming message delivery.
OracleEBSPaletteConnectionNotFoundException	The connection cannot be created.
OracleEBSPaletteSQLException	An SQL Exception occurs.

## Custom Concurrent Program

You can use the Custom Concurrent Program activity to call custom concurrent programs written to integrate with Oracle E-Business Suite that are not registered with Oracle Integration Repository.

For the Custom Concurrent Program activity, you must specify more information about the concurrent program and interface tables. After providing the information, you can easily configure the plug-in activity through the user interface.



A comparison between Oracle Concurrent Program and Custom Concurrent Program is provided in [Oracle Concurrent Program and Custom Concurrent Program Comparison](#). If you create a Custom Concurrent Program and register it into the Integration Repository, it becomes a regular Oracle Concurrent Program and is treated as such.

The Custom Concurrent Program activity provides the following functions:

- At design time
  - Loads Concurrent Program parameters from the FND table.
  - Loads Inbound Interface Tables or Views from the database manually.
  - Saves the information to a file.
- At run time
  - Loads Concurrent Program parameters and Interface Tables or Views from the file.
  - Uses JDBC to insert data into Interface Tables or Views.
  - Uses JDBC to submit a Concurrent Program request.





Custom Concurrent Program can be used in Oracle E-Business Suite 12.0.x, 12.1.x, and 12.2.x.



### General

In the **General** tab, you can establish a connection to an Oracle E-Business Suite server, specify the language, concurrent program name, responsibility name, and user name, and set the check interval and total amount of time for checking the result of executing the Concurrent Program.

The **General** tab contains the following fields:

Field	Module Property?	Description
<b>Name</b>	No	The name of the resource. The default is CustomConcurrentProgram.

Field	Module Property?	Description
<b>Shared Connection</b>	Yes	<p>A shared configuration resource containing the Oracle E-Business Suite connection information.</p> <p>To select a shared connection resource, click the <b>Choose/Create Default Resource</b>  icon. All usable connections are listed in the opened dialog. Select one to apply to your current activity. If no matching connection resource is found, click <b>Create Shared Resource</b> to create one. See <a href="#">Oracle E-Business Suite Connection</a> for more details.</p> <p> Before running the Custom Concurrent Program activity, ensure that the plug-in user name that you specified in the Shared Connection is correct.</p>
<b>Language</b>	No	<p>The language in which the Oracle E-Business Suite user interface is shown.</p> <p>Click <b>Search Language</b> or the text field. Then, enter a part of the language name in the opened dialog or leave the text field empty, and click <b>Search</b>. All languages that are installed in Oracle E-Business Suite and match the search criteria are listed in the dialog. Select a language in the opened dialog, and click <b>OK</b>. The contents of the concurrent program name, responsibility name, and output change accordingly.</p>
<b>Concurrent Program Name</b>	No	<p>Programs that can be called by a concurrent manager.</p> <p>Click <b>Search Concurrent Program Name</b> or the text field. Then, enter a part of the concurrent program name in the opened dialog or leave the text field empty, and click <b>Search</b>. Select a Concurrent Program from the Concurrent Program List dialog, and click <b>OK</b>.</p>
<b>Responsibility Name</b>	No	<p>A responsibility is a level of authority in Oracle E-Business Suite. You can only access the functions and data specified by the selected responsibility.</p> <p>Click <b>Search Responsibility Name</b> or the text field. Then, enter a part of the responsibility name in the opened dialog or leave the text field empty, and click <b>Search</b>. Select a Concurrent Program from the Concurrent Program List dialog, and click <b>OK</b>.</p>
<b>User Name</b>	No	<p>The name of the user whose information is stored in Oracle E-Business Suite.</p> <p>Click <b>Search User Name</b> or the text field. Then, enter a part of the user name in the opened dialog or leave the text field empty, and click <b>Search</b>. Select a user name from the User Name List dialog, and click <b>OK</b>.</p>
<b>Concurrent Request</b>	No	<p>If this check box is selected, the plug-in inserts records into interface tables, and submits the concurrent request. Otherwise, the plug-in only inserts records into interface tables.</p>

Field	Module Property?	Description
<b>Wait for Response</b>	No	<p>Select this check box if you want the Concurrent Program to retrieve the <b>REQUEST_ID</b>, <b>PHASE</b>, <b>STATUS</b>, and <b>MESSAGE</b> information, which are shown in the <b>Output</b> tab. If this check box is cleared, the response time becomes shorter, but only <b>REQUEST_ID</b> is retrieved.</p>
<b>Check Interval (s)</b>	Yes	<p>The time interval for checking the result of executing the Concurrent Program.</p> <p>The default is 15.</p> <div style="border-left: 1px solid #ccc; padding-left: 10px; margin-top: 10px;">  <p>When running the Custom Concurrent Program activity, if the total waiting time that you set for checking the result of the Concurrent Program activity is shorter than the check interval, the plug-in still checks the result once.</p> </div>
<b>Wait Time (s)</b>	Yes	<p>The total amount of time allowed for checking the result of executing the Concurrent Program.</p> <p>The time set in this field should be a multiple of the time set in the <b>Check Interval (s)</b> field. The default is 0, which means the plug-in waits until the Concurrent Program execution is completed.</p>
<b>Interface Table Configuration</b>	No	<p>Click this button to add the interface tables or views (if they exist) related to the Concurrent Program.</p> <div style="border-left: 1px solid #ccc; padding-left: 10px; margin-top: 10px;">  <p>Different with Oracle Concurrent Program, the Integration Repository does not provide the relevant information of the interface tables for Custom Concurrent Program. You can get the relevant information of the interface tables for Custom Concurrent Program from custom defined interface tables configured in the Interface Table Configuration dialog, or from the Oracle documentation and Oracle Support.</p> </div>

Field	Module Property?	Description
<b>Generate SQL Script</b>	No	<p>After configuring the Custom Concurrent Program activity, click <b>Generate SQL Script</b>. The scripts to be generated are displayed in the opened dialog. Click <b>Generate</b>, and the following two SQL scripts are created in the <code>TIBCO_HOME\bw\palettes\oracleebbs\version_number\sql</code> directory:</p> <ul style="list-style-type: none"> <li>• <code>ProcessName_ActivityName_ConcurrentProgramName.sql</code></li> <li>• <code>ProcessName_ActivityName_ConcurrentProgramName_undo.sql</code></li> </ul> <p>Run the <code>ProcessName_ActivityName_ConcurrentProgramName.sql</code> script in the connected Oracle database before starting the process.</p> <p>The <code>ProcessName_ActivityName_ConcurrentProgramName.sql</code> script is used to grant privileges to the plug-in user so that the plug-in user can access the objects being configured. The plug-in uses the apps user entered in the Oracle E-Business Suite Connection resource to get the information to generate this script. After this script is executed, the plug-in user can access the tables and procedures configured for the activity.</p> <p>The script with the <code>_undo</code> suffix reverses the granted privileges and undoes the changes to the database. With this operation, you can cleanse the privileges for objects that are no longer used by obsolete or unwanted activities.</p>



If the execution of the Concurrent Program takes a significant amount of time, you can clear the **Wait for Response** check box, and just get `REQUEST_ID` in the **Output** tab. Or you select the **Wait for Response** check box, and set the **Wait Time (s)** field to a limited time; in this case, the Concurrent Program returns the output data no matter whether the execution of the Concurrent Program is completed or not.

### Description

In the **Description** tab, you can enter a short description for the Custom Concurrent Program activity.

### Concurrent Program Details

In the **Concurrent Program Details** tab, you can view the `FND_REQUEST.OPTIONS` and `FND_REQUEST.SUBMIT_REQUEST` procedures in the plug-in user schema and the related interface tables or views (if they exist).

### Input

In the **Input** tab, you can view the input data for the Custom Concurrent Program activity. The parameters in the `CustomConcurrentProgram-input` column correspond to those shown in the **Concurrent Program Details** tab, with the following exceptions: **application**, **program**, **description**, **start\_time**, and **sub\_request**.



Select the **Concurrent Request** check box in the **General** tab to show FND\_REQUEST.SUBMIT\_REQUEST. The parameters of the FND\_REQUEST.SUBMIT\_REQUEST procedure, **argument1 - 100**, are listed without showing the name of each parameter or whether the parameter is required. This is the standard implementation for the FND\_REQUEST.SUBMIT\_REQUEST procedure. You can find the real arguments by the scripts in [Running the CustomCP Project](#). Also, for non-SRS Concurrent Programs, you must use this standard implementation. See [Oracle Concurrent Program and Custom Concurrent Program Comparison](#) for more details.

The **ResponsibilityName** and **UserName** are dynamic parameters. They are configured in the **General** tab. You can also change the **ResponsibilityName** and **UserName** parameters in the **Input** tab at run time.



The parameter settings in the **Input** tab takes precedence over those in the **General** tab.

### Output

In the **Output** tab, you can find the output information for the Custom Concurrent Program activity. If the **Wait for Response** check box is selected in the **General** tab, the **REQUEST\_ID**, **PHASE**, **STATUS**, and **MESSAGE** information are all displayed in this tab. Otherwise, only **REQUEST\_ID** is retrieved.



If you clear the **Concurrent Request** check box in the **General** tab, the output is empty.

### Fault

In the **Fault** tab, you can find the error messages of the Custom Concurrent Program activity.

See [Plug-in Error Codes](#) for more information about error messages and corrective actions to take.

The **Fault** tab lists the following error messages:

Fault	Thrown When ...
OracleEBSPaletteAQConnectionException	The AQ Connection does not initialize. A JMSEException happens during an incoming message delivery.
OracleEBSPaletteConnectionNotFoundException	The connection cannot be created.
OracleEBSPaletteSQLException	An SQL Exception occurs.

## Oracle Business Event

You can use the Oracle Business Event activity to listen for an Oracle Business Event triggered in the Oracle E-Business Suite application.

When the Oracle Business Event is triggered, the Oracle Business Event activity pulls the event data from the Oracle event queue, and makes the data available to TIBCO environment. Custom business events configured in Oracle E-Business Suite and registered into Oracle Integration Repository are compatible with this activity.

A business event represents an action or occurrence triggered by a business process in an application or system. This event might be significant to other programs, applications, processes, or external agents. In Oracle E-Business Suite, events can be triggered by business object manipulation such as an update or change, or by different processes such as the approval of a purchase order.




The Oracle Business Event activity provides the following functions:

- At design time
  - Generates PL/SQL scripts to help users subscribe a Business Event to a specified Oracle Advance Queue (AQ).
  - Connects to the AQ, retrieves an event, and parses the event schema.
- At run time
  - Connects to the AQ and listens to the message (event) from AQ.
  - Parses the event content.

## General

In the **General** tab, you can establish a connection to an Oracle E-Business Suite server, and specify the event name.

The **General** tab contains the following fields:

Field	Module Property?	Description
<b>Name</b>	No	The name of the resource. The default is OracleBusinessEvent.
<b>Shared Connection</b>	Yes	<p>A shared configuration resource containing the Oracle E-Business Suite connection information.</p> <p>To select a shared connection resource, click the <b>Choose/Create Default Resource</b>  icon. All usable connections are listed in the opened dialog. Select one to apply to your current activity. If no matching connection resource is found, click <b>Create Shared Resource</b> to create one. See <a href="#">Oracle E-Business Suite Connection</a> for more details.</p> <p> Before running the Oracle Business Event activity, ensure that both the apps user name and plug-in user name that you specified in the Shared Connection are correct.</p>
<b>Event Name</b>	No	<p>The name of the Business Event. For example, <code>oracle.apps.per.api.person.update_person</code>.</p> <p>Click <b>Search Event</b> or the text field. Then, enter a part of the event name in the opened dialog or leave the text field empty, and click <b>Search</b>. Select an Oracle Business Event from the Oracle Business Event List dialog, and click <b>OK</b>.</p> <p>The Oracle Business Event List dialog lists two types of Business Events, Event and Group. Only the Event type Business Events are supported in this release.</p> <p> When you select an Event Name from the list, certain background operations are executed. The Business Event can be checked in Oracle E-Business Suite.</p>

Field	Module Property?	Description
<b>Generate SQL Script</b>	No	<p>After configuring the Oracle Business Event activity, click <b>Generate SQL Script</b>. The scripts to be generated are displayed in the opened dialog. Click <b>Generate</b>, and the following two SQL scripts are created in the <code>TIBCO_HOME\bw\palettes\oracleeb\version_number\sql</code> directory:</p> <ul style="list-style-type: none"> <li>• <code>ProcessName_ActivityName_EventName.sql</code></li> <li>• <code>ProcessName_ActivityName_EventName_undo.sql</code></li> </ul> <p>Run the <code>ProcessName_ActivityName_EventName.sql</code> script in the connected Oracle database before starting the process.</p> <p>The <code>ProcessName_ActivityName_EventName.sql</code> script creates the necessary infrastructure that makes the event available for use by the plug-in Oracle Business Event activity without having to manually configure this event in Oracle E-Business Suite. This involves the creation of a subscription, an agent, and the required linking of the Oracle business event to these entities.</p> <p>The script with the <code>_undo</code> suffix reverses the granted privileges and undoes the changes to the database. With this operation, you can cleanse the privileges for objects that are no longer used by obsolete or unwanted activities.</p>

### Description

In the **Description** tab, you can enter a short description for the Oracle Business Event activity.

### Event Attributes

In the **Event Attributes** tab, you can view the attributes of the Oracle Business Event.

Click **Get Event Attributes**. The attributes of the Business Event are parsed from XML format and displayed in this tab.



When you click **Get Event Attributes**, the event attributes are also displayed in the **Output** tab in XML format.

You can also click **Delete Event Attributes** to delete the event attributes. In this case, the event attributes in the **Output** tab are also removed.

### Advanced

In the **Advanced** tab, you can specify the agent name, queue name, and queue table name, validate the configuration, and specify the sequence key and custom job ID. In addition, if multiple consumers are involved in the activity, you must specify the subscriber name.

The **Advanced** tab contains the following fields:



Field	Module Property?	Description
<b>Select an existing agent?</b>	No	Select this check box to use an existing agent. If this check box is selected, the <b>Agent Name</b> , <b>Queue Name</b> , <b>Queue Table Name</b> , <b>Multiple Consumer</b> , and <b>Subscriber Name</b> fields are disabled, and the <b>Select Agent</b> button is applicable.
<b>Agent Name</b>	No	Enter a name for the agent that you intend to create to receive messages from Oracle Business Event System.
<b>Queue Name</b>	No	Enter a name for the queue to receive messages from the agent.
<b>Queue Table Name</b>	No	Enter a name for the queue table.
<b>Multiple Consumer</b>	No	Select this check box, if you want the messages to be consumed by multiple subscribers. In this case, the messages are propagated by using the topic method. Otherwise, the messages are propagated by using the queue method.
<b>Subscriber Name</b>	No	Enter the name of the subscriber who consumes the messages, or click <b>Select Subscriber</b> to select an existing subscriber. This field is applicable only when the <b>Multiple Consumer</b> check box is selected.
<b>Validate Configuration</b>	No	Validate the configuration of the <b>Agent Name</b> , <b>Queue Name</b> , and <b>Queue Table Name</b> fields.
<b>Sequence Key</b>	No	This field contains an XPath expression that specifies the order in which the process run. Process instances with sequencing keys that have the same value are executed sequentially in the order in which the process instances are created.  See the TIBCO ActiveMatrix BusinessWorks documentation for more information.
<b>Custom Job Id</b>	No	This field contains an XPath expression that specifies a custom job ID for the process instance. This ID is displayed in the TIBCO Administrator View Service dialog, and it is also available in the <b>\$_processContext</b> process variable.  See the TIBCO ActiveMatrix BusinessWorks documentation for more information.



When specifying the agent name, queue name, queue table name, and subscriber name, the texts entered are capitalized automatically.

When a Business Event is triggered in Oracle E-Business Suite, the plug-in listens to the event.

- If you want the event messages to be received by a single consumer, create a new set of agent, queue, and queue table for the subscription, or use an existing set of agent, queue, and queue table for the same Business Event.
  - To create a new set of agent, queue, and queue table, clear the **Select an existing agent** check box, and enter the names of the agent, queue, and queue table in their respective fields, as

shown in the following figure. The agent, queue, and queue table are created in Oracle E-Business Suite after you run the generated SQL script.

The screenshot shows the configuration window for an OracleBusinessEvent. The 'General' tab is selected. The 'Select an existing agent?' checkbox is unchecked. The 'Agent Name' field contains 'TIB\_WF\_AGENT', the 'Queue Name' field contains 'TIB\_WF\_Q', and the 'Queue Table Name' field contains 'TIB\_WF\_QTAB'. The 'Multiple Consumer' checkbox is unchecked, and the 'Subscriber Name' field is empty. The 'Validate Configuration' button is visible.

- To use an existing set of agent, queue, and queue table, select the **Select an existing agent** check box. In this case, the **Agent Name**, **Queue Name**, **Queue Table Name**, **Multiple Consumer**, and **Subscriber Name** fields are disabled, and the **Select Agent** button is applicable. Click **Select Agent** to select an existing agent for the subscription, and the related queue and queue table are obtained simultaneously, as shown in the following figure.

The screenshot shows the configuration window for an OracleBusinessEvent. The 'Select an existing agent?' checkbox is checked. The 'Agent Name' field contains 'TIB\_WF\_AGENT', the 'Queue Name' field contains 'TIB\_WF\_Q', and the 'Queue Table Name' field contains 'TIB\_WF\_QTAB'. The 'Multiple Consumer' checkbox is unchecked, and the 'Subscriber Name' field is empty. The 'Validate Configuration' button is visible.

- If you want the event messages to be received by multiple consumers, first, create a set of agent, queue, and queue table for the subscription of event messages; then specify the subscriber name.
  - To create a new set of agent, queue, and queue table for the subscription of event messages to be received by multiple consumers, clear the **Select an existing agent** check box, and specify the names of the agent, queue, and queue table in their respective fields. Select the **Multiple Consumer** check box, and specify the name of the subscriber that receives the event messages in the specific configuration, as shown in the following figure. The agent, queue, queue table, and subscriber are created in Oracle E-Business Suite after you run the generated SQL script.

The screenshot shows the configuration window for an OracleBusinessEvent. The 'Select an existing agent?' checkbox is unchecked. The 'Agent Name' field contains 'TIB\_WF\_AGENT\_PERSON1', the 'Queue Name' field contains 'TIB\_WF\_QUEUE\_PERSON1', and the 'Queue Table Name' field contains 'TIB\_WF\_QTAB\_PERSON1'. The 'Multiple Consumer' checkbox is checked, and the 'Subscriber Name' field contains 'TIB\_WF\_SUB\_PERSON1'. The 'Validate Configuration' button is visible.

- To use this set of agent, queue, and queue table in other configurations, for example, in the following figure, select the **Select an existing agent** check box, and then select this agent. The related queue and queue table are obtained simultaneously; the **Multiple Consumer** check box is selected automatically and the **Subscriber Name** field is enabled. Specify the name of the subscriber. The subscriber is created in Oracle E-Business Suite after you run the generated SQL script. This subscriber can then receive messages from the same set of agent, queue, and queue table.

The screenshot shows the configuration interface for an Oracle Business Event. The 'Advanced' tab is active, displaying the following settings:

- Select an existing agent? :**
- Agent Name:** TIB\_WF\_AGNET\_PERSON1 (with a 'Select Agent' button)
- Queue Name:** TIB\_WF\_QUEUE\_PERSON1
- Queue Table Name:** TIB\_WF\_QTAB\_PERSON1
- Multiple Consumer:**
- Subscriber Name:** TIB\_WF\_SUB\_PERSON2 (with a 'Select Subscriber' button)
- Validate Configuration:** [Validate Configuration button]
- Sequence Key:** [Empty field]
- Custom Job Id:** [Empty field]

## Conversations

In the **Conversations** tab, you can initiate a conversation. In addition, you can click **Add New Conversation** to initiate multiple conversations.

See *TIBCO ActiveMatrix BusinessWorks Samples* for details about conversations.

## Output

In the **Output** tab, you can find the output information for the Oracle Business Event activity.

## Working with Sample Projects

---

After installing the plug-in, you can locate the sample projects in the `TIBCO_HOME\bw\palettes\oracleebs\version_number\examples` directory.

The plug-in packages five sample projects with the installer. The sample projects show how to use the activities of TIBCO ActiveMatrix BusinessWorks Plug-in for Oracle E-Business Suite in dealing with inbound and outbound business objects between TIBCO environment and Oracle E-Business Suite.

This plug-in provides the following sample projects:

- [Working with the OracleAPI Project](#)
- [Working with the CustomAPI Project](#)
- [Working with the OracleCP Project](#)
- [Working with the CustomCP Project](#)
- [Working with the BusinessEvent Project](#)

### Importing Sample Projects

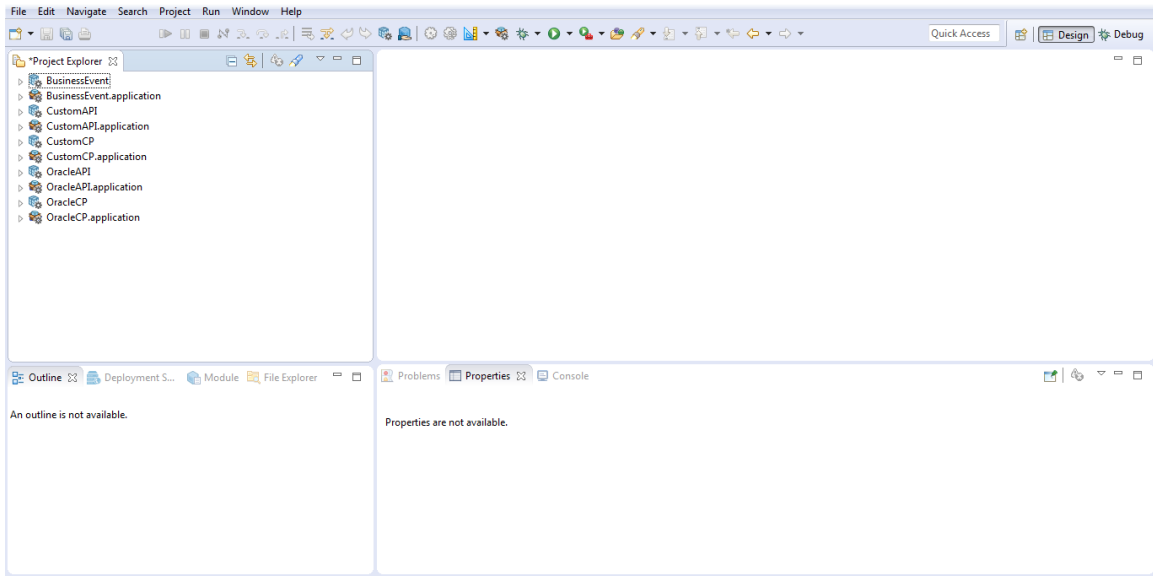
Before running the projects, you must import the projects to TIBCO Business Studio.

#### Procedure

1. Start TIBCO Business Studio by using one of the following ways:
  - Microsoft Windows: click **Start > All Programs > TIBCO > TIBCO\_HOME > TIBCO Business Studio version\_number > Studio for Designers**.
  - Mac OS and Linux: run the TIBCO Business Studio executable file located in the `TIBCO_HOME/studio/version_number/eclipse` directory.
2. From the menu, click **File > Import**.
3. In the Import dialog, expand the **General** folder, and select **Existing Studio Projects into Workspace**. Click **Next**.
4. Click **Browse** next to the **Select root directory** field to locate the sample projects. Click **Finish**.  
The sample projects are located in the `TIBCO_HOME\bw\palettes\oracleebs\6.1\examples` directory.

#### Result

The sample projects are imported to TIBCO Business Studio, as shown in the following figure.



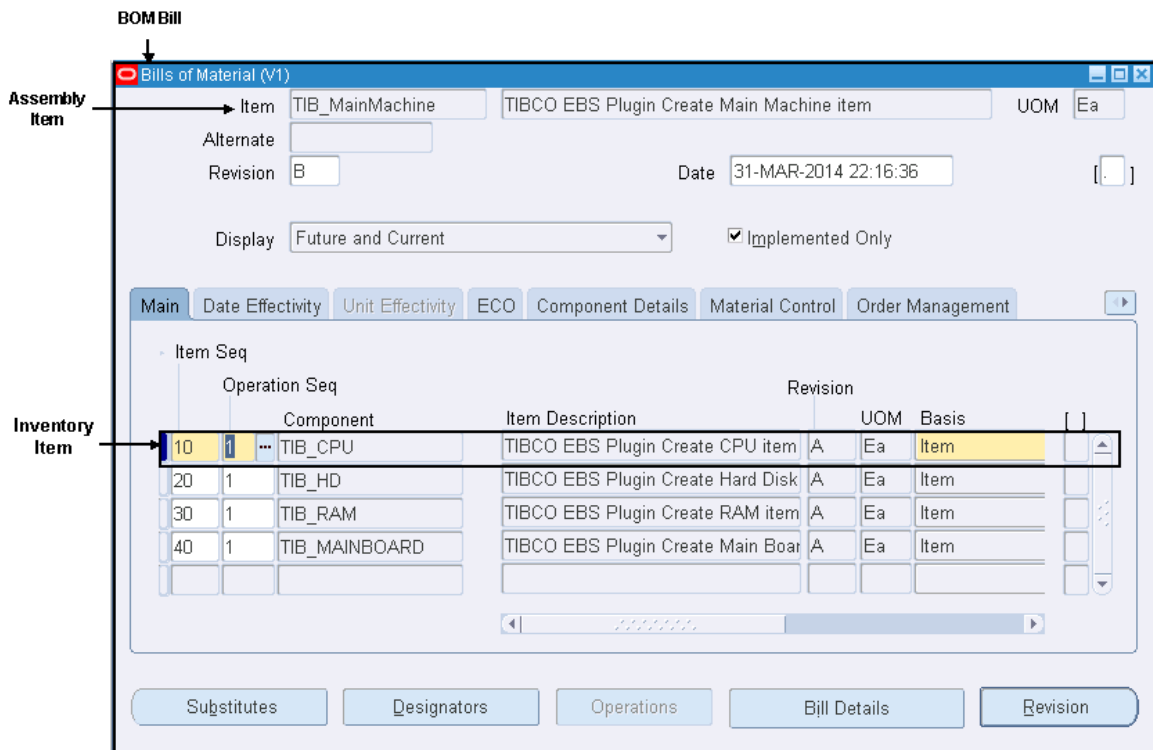
## Working with the OracleAPI Project

The OracleAPI project demonstrates how to create inventory items and BOM (Bill of Material) bills by Oracle PL/SQL Interface through the TIBCO ActiveMatrix BusinessWorks Plug-in for Oracle E-Business Suite.



This example requires the use of the Vision Demo environment. It relies on responsibility, users, and data in the demo installation.

Two processes are preconfigured in the OracleAPI project, [EGO\\_ITEM\\_PUB.bwp process](#) and [BOM\\_BO\\_PUB.bwp process](#). In these two processes, inventory items, assembly item, and BOM bill are created, as shown in the following figure.



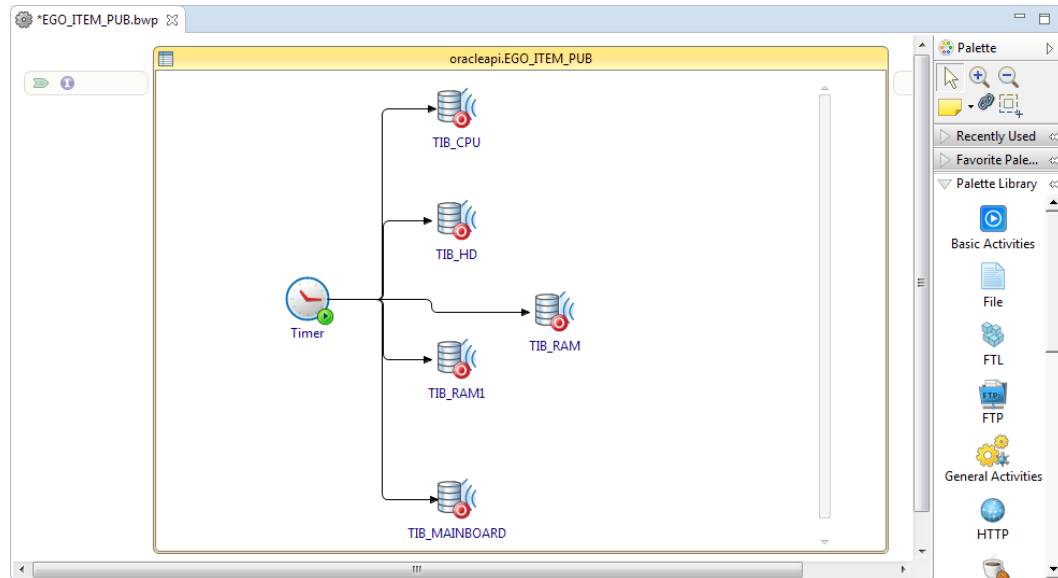


This example is configured to work against the Oracle E-Business Suite Vision Demo environment. All inputs are valid under this environment.

### EGO\_ITEM\_PUB.bwp Process

The EGO\_ITEM\_PUB.bwp process creates inventory items for Oracle E-Business Suite by using the Oracle API activities.

The process is designed with the following activities:



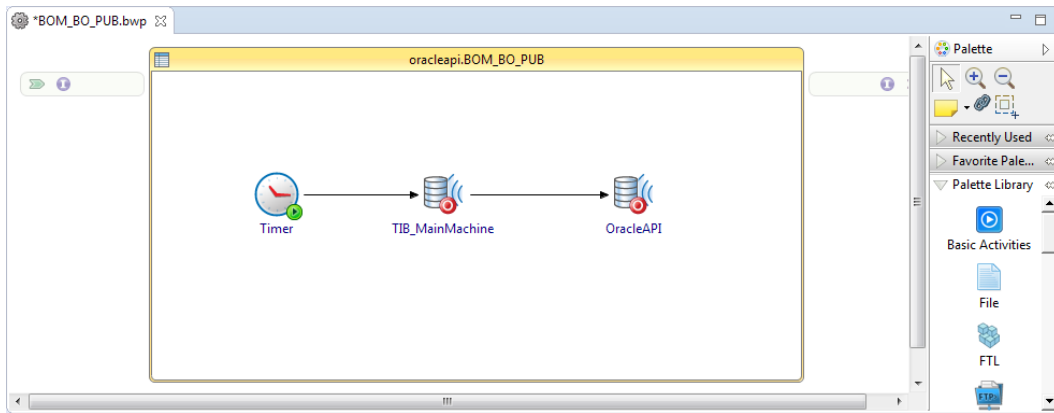
For the five activities in this process, you can create an activity, and then rename the activity and modify the parameters to create different activities and run the activities separately; or you can create all activities at a time and run them simultaneously.

Activity	Description
TIB_CPU	Creates the CPU inventory item for Oracle E-Business Suite.
TIB_HD	Creates the Hard Disk inventory item for Oracle E-Business Suite.
TIB_RAM	Creates the RAM inventory item for Oracle E-Business Suite.
TIB_RAM1	Creates the RAM1 inventory item for Oracle E-Business Suite.
TIB_MAINBOARD	Creates the Main Board inventory item for Oracle E-Business Suite.

### BOM\_BO\_PUB.bwp Process

The BOM\_BO\_PUB.bwp process creates an assembly item and a BOM bill for Oracle E-Business Suite by using the Oracle API activities.

The process is designed with the following activities:



Activity	Description
TIB_MainMachine	Creates the Main Machine assembly item for Oracle E-Business Suite.
OracleAPI	Creates a BOM bill for Oracle E-Business Suite.

## Running the EGO\_ITEM\_PUB.bwp Process

You can run the EGO\_ITEM\_PUB.bwp process in the OracleAPI project to see how to create inventory items for Oracle E-Business Suite by using the Oracle API activities.



See [EGO\\_ITEM\\_PUB Package Configuration](#) for details of the configurations and inputs of the activities in the EGO\_ITEM\_PUB.bwp process.

### Prerequisites

Before running the project, ensure that you have connected to an Oracle Database Server and imported the project to TIBCO Business Studio. See [Connecting to Oracle Database Server](#) and [Importing Sample Projects](#) for more details.

### Procedure





1. In the Project Explorer view, expand the OracleAPI project.
2. Configure the Oracle E-Business Suite connection:
  - a) Expand **Resources** > **oracleapi**.
  - b) Double-click `OEBSConnectionResource.oebconnectionResource`.
  - c) In the OEBS Connection editor, edit the resource connection, and then click **Test Connection** to validate your connection.

See [Oracle E-Business Suite Connection](#) for more details about how to configure the connection.

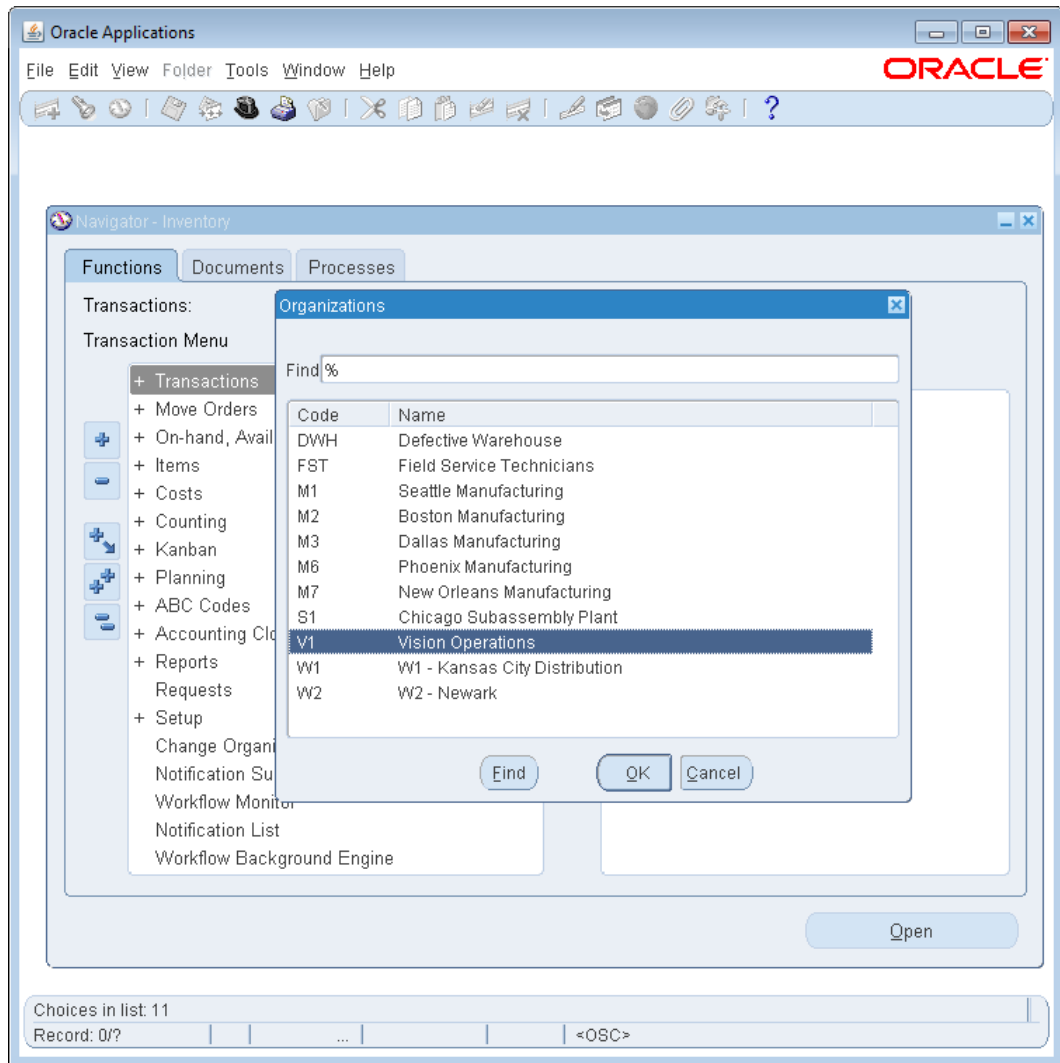
3. Open the process:
  - a) Expand **Processes** > **oracleapi**.
  - b) Double-click `EGO_ITEM_PUB.bwp`.



The activities in this process are already configured; therefore, you do not need to configure the activities. However, if you switch to a different Oracle E-Business Suite version, you must click **Search Procedure** to reselect the PROCESS\_ITEM procedure. This is because the parameter number of the PROCESS\_ITEM procedure in the EGO\_ITEM\_PUB package changes when you connect to a different version of Oracle E-Business Suite. In Oracle E-Business Suite 12.1.x and 12.0.x, the parameter number of the PROCESS\_ITEM procedure is 74; in Oracle E-Business Suite 12.2.x, the parameter number is 75. The samples of plug-in 6.1.0 are created on Oracle E-Business Suite 12.1.x.

4. Expand **Module Descriptors** and double-click **Components**.  
By default, both processes are listed in the Components editor. Click the **Create Process Component**  icon to add the process you want to run, or click  to remove the process.
5. Save the project.
6. From the menu, Click **Run > Debug Configurations**, or click  **Debug > Debug Configurations**.
7. Click **BusinessWorks Application > BWApplication** in the left panel in the Debug Configurations window.  
By default, all applications in the current workspace are selected in the **Applications** tab.
8. Click **Deselect All** in the **Applications** tab, and select the check box next to the project you want to run.
9. Click **Debug** to run the process.
10. Click the **Terminate**  icon in the Console view to stop the running process.  
You can check the value of **X\_RETURN\_STATUS** in the **Output** tab in the Job Data view to see whether the process runs successfully. "s" stands for success, and "E" stands for error.
11. Request result in Oracle E-Business Suite:
  - a) Log in to the Oracle E-Business Suite system with the user name MFG and password welcome.
  - b) Click **Inventory > Items > Master Items** to open the Oracle Applications window with the Navigator - Inventory dialog and the Organizations dialog is displayed.

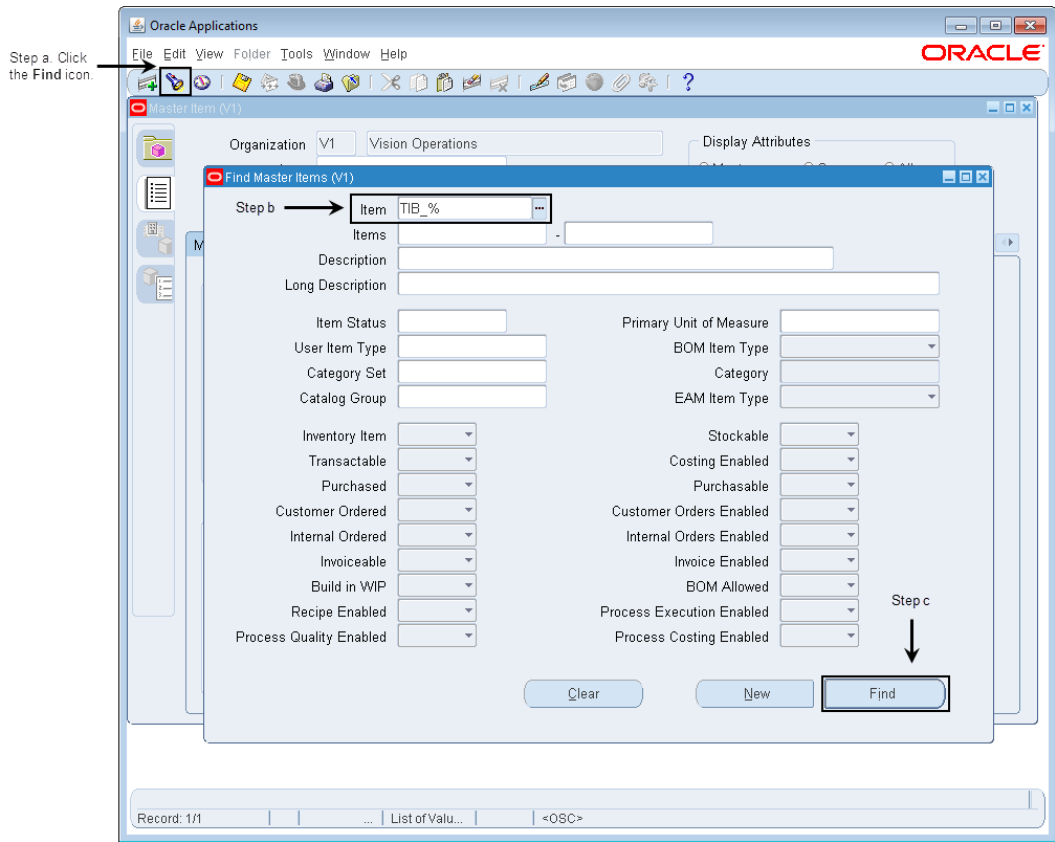




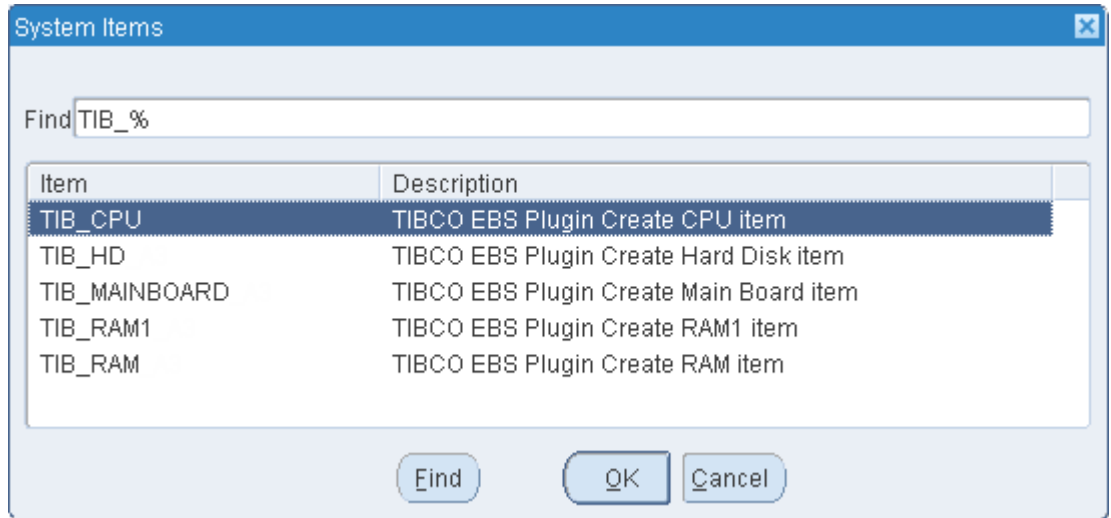
c) Select V1 organization from the Organizations dialog, and click **OK**.

d) Click the **Find**  icon.

e) In the Find Master Items (V1) dialog, specify the search filter in the **Item** field (for example, TIB\_%), and click **Find**.



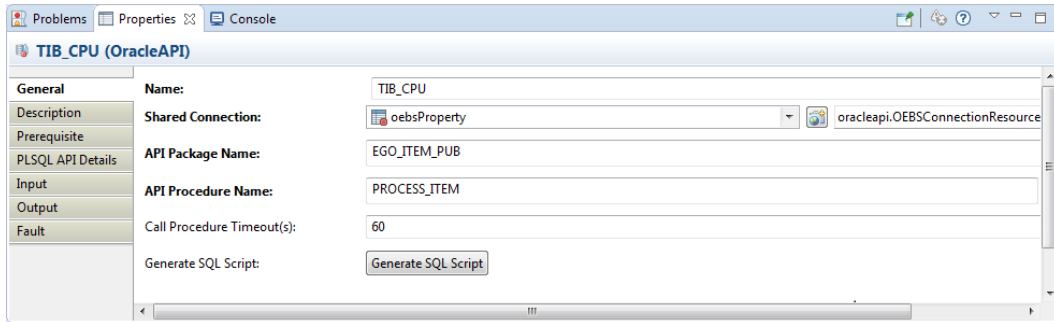
The following figure shows the result in the System Items dialog:



### EGO\_ITEM\_PUB Package Configuration

You can use the EGO\_ITEM\_PUB package to create inventory items and assembly items.

The following figure shows a sample configuration of the EGO\_ITEM\_PUB package:



As defined in Oracle Integration Repository, EGO\_ITEM\_PUB.PROCESS\_ITEM is a convenient wrapper to Process\_Item. You can use this API to create or update an item by passing only the most important and commonly used item attributes.

The following table lists the EGO\_ITEM\_PUB package inputs of the TIB\_CPU activity in the OracleAPI project:

Parameter	Type	Input
P_API_VERSION	NUMBER	1.0
P_INIT_MSG_LIST	VARCHAR2	T
P_COMMIT	VARCHAR2	T
P_TRANSACTION_TYPE	VARCHAR2	CREATE
P_LANGUAGE_CODE	VARCHAR2	US
P_TEMPLATE_ID	NUMBER	259
P_TEMPLATE_NAME	VARCHAR2	Purchased Item
P_SEGMENT1	VARCHAR2	TIB_CPU
P_ORGANIZATION_ID	NUMBER	204
P_ORGANIZATION_CODE	VARCHAR2	V1
P_CATALOG_STATUS_FLAG	VARCHAR2	N
P_DESCRIPTION	VARCHAR2	TIBCO EBS Plug-in Create CPU item
P_PRIMARY_UOM_CODE	VARCHAR2	Ea
P_INVENTORY_ITEM_STATUS_CODE	VARCHAR2	Active
P_BOM_ENABLED_FLAG	VARCHAR2	Y
P_ENG_ITEM_FLAG	VARCHAR2	N



You can define these parameters manually or automatically by defining a module property. For how to define module properties, see "Using Process and Module Properties" in *TIBCO ActiveMatrix BusinessWorks Samples*.

## Running the BOM\_BO\_PUB.bwp Process

You can run the BOM\_BO\_PUB.bwp process in the OracleAPI project to see how to create assembly items and BOM bills for Oracle E-Business Suite by using the Oracle API activities.



See [EGO\\_ITEM\\_PUB Package Configuration](#) and [BOM\\_BO\\_PUB Package Configuration](#) for details of the configurations and inputs of the activities in the BOM\_BO\_PUB.bwp process.

### Prerequisites

Run the EGO\_ITEM\_PUB.bwp process successfully. See [Running the EGO\\_ITEM\\_PUB.bwp Process](#) for details.

### Procedure

1. Open the process:
  - a) Expand **Processes > oracleapi**.
  - b) Double-click `BOM_BO_PUB.bwp`.






The activities in the process are already configured; therefore, you do not need to configure the activities. However, if you switch to a different Oracle E-Business Suite version, for the TIB\_MainMachine activity, you must click **Search Procedure** to reselect the PROCESS\_ITEM procedure. This is because the parameter number of the PROCESS\_ITEM procedure in the EGO\_ITEM\_PUB package changes when you connect to a different version of Oracle E-Business Suite. In Oracle E-Business Suite 12.1.x and 12.0.x, the parameter number of the PROCESS\_ITEM procedure is 74; in Oracle E-Business Suite 12.2.x, the parameter number is 75. The samples of plug-in 6.1.0 are created on Oracle E-Business Suite 12.1.x.


2. Execute the `BOM_BO_PUB.sql` file by using the apps user on the command line.

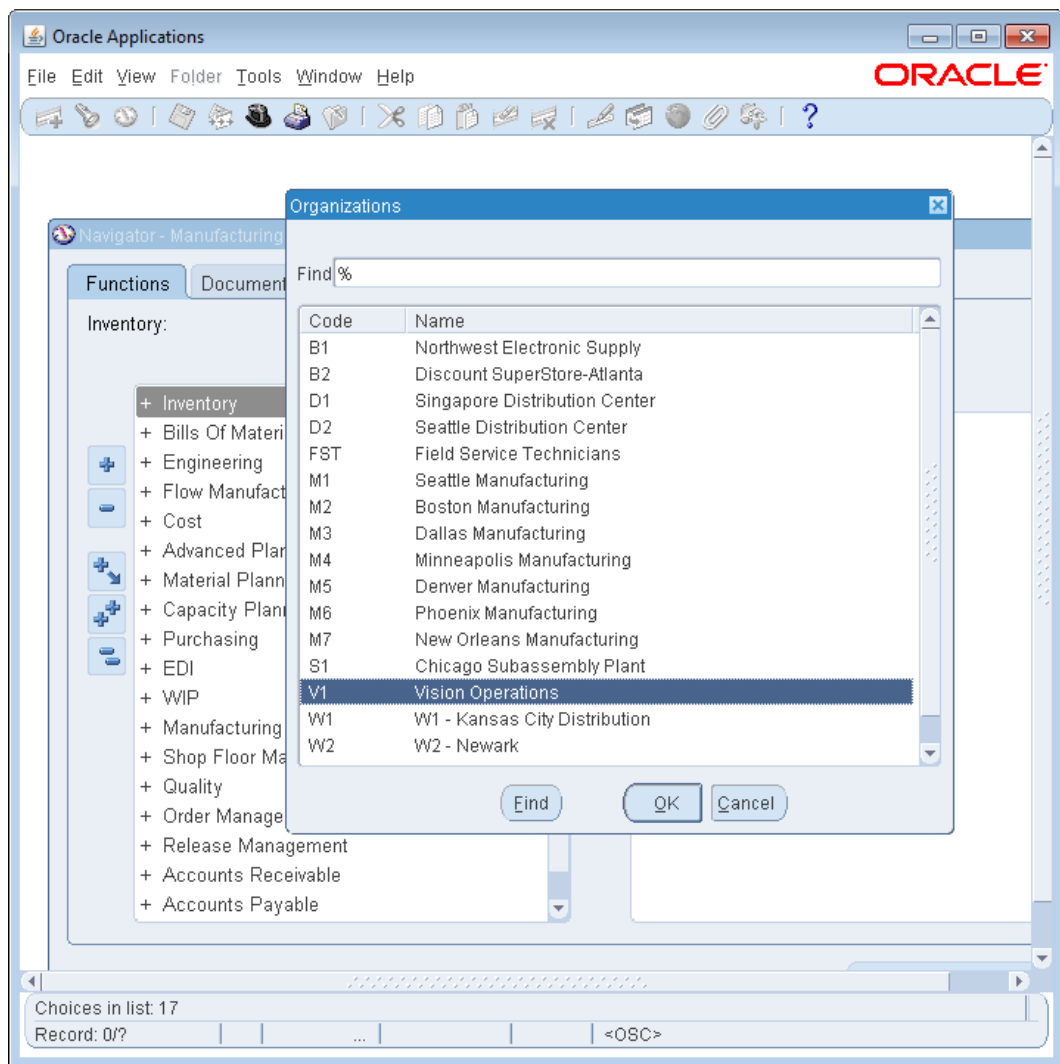
Because the BOM\_BO\_PUB.PROCESS\_BOM procedure contains package types, you must use JPublisher to generate wrapper package first. In this example, two PL/SQL files are generated, `BOM_BO_PUB.sql` and `BOM_BO_PUB_drop.sql`. See [Introduction to Oracle Database JPublisher](#) for more details.



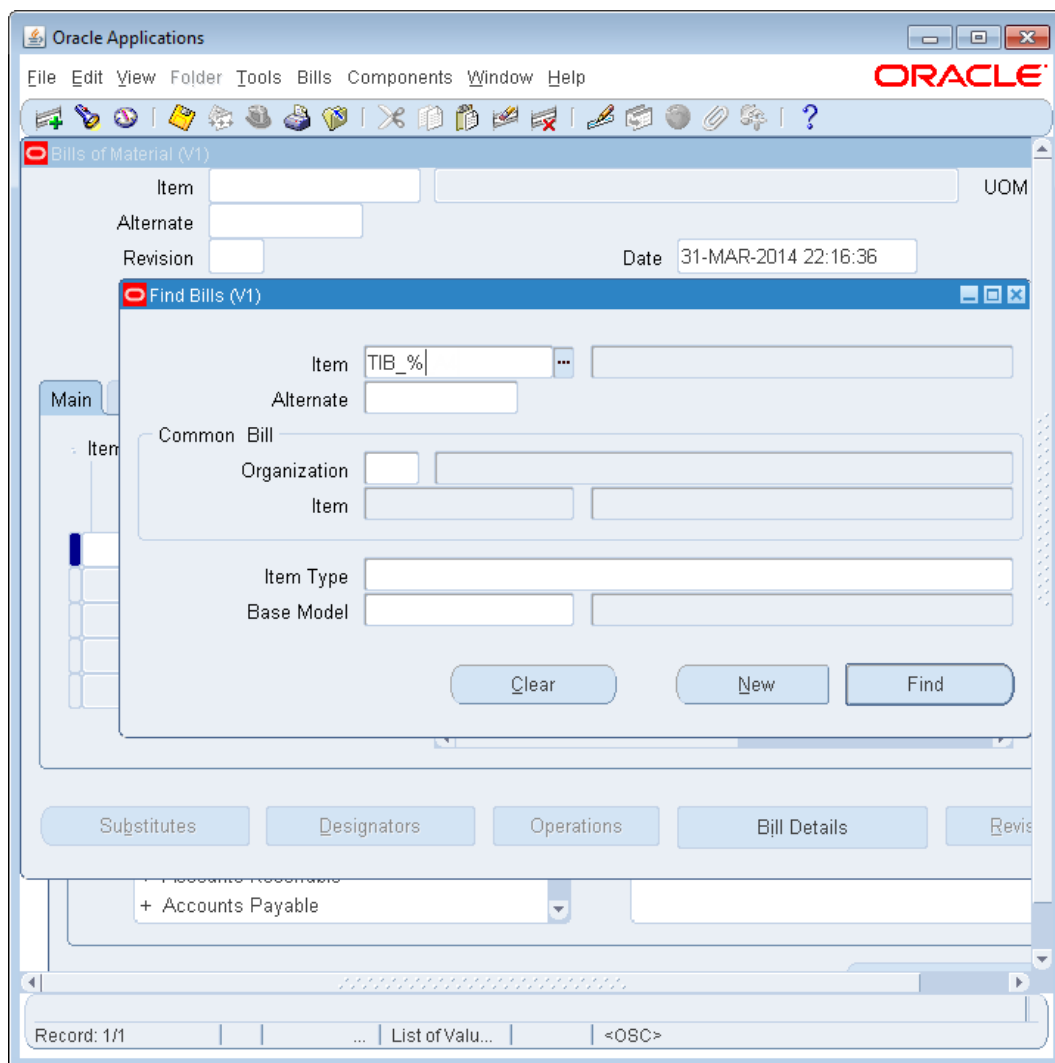
After using JPublisher to generate the wrapper package for the BOM\_BO\_PUB.PROCESS\_BOM procedure, if you issue the `BOM_BO_PUB.sql` file, issue the `BOM_BO_PUB_drop.sql` file, reissue the `BOM_BO_PUB.sql` file, and then run the process, the `invalid data type error` exception is thrown. To solve this issue, you can restart TIBCO Business studio, and then run the process again.

3. Expand **Module Descriptors** and double-click **Components**.  
By default, both processes are listed in the Components editor. Click the **Create Process Component**  icon to add the process you want to run, or click  to remove the process.
4. Save the project.
5. From the menu, Click **Run > Debug Configurations**, or click  **Debug > Debug Configurations**.
6. Click **BusinessWorks Application > BApplication** in the left panel in the Debug Configurations window.  
By default, all applications in the current workspace are selected in the **Applications** tab.

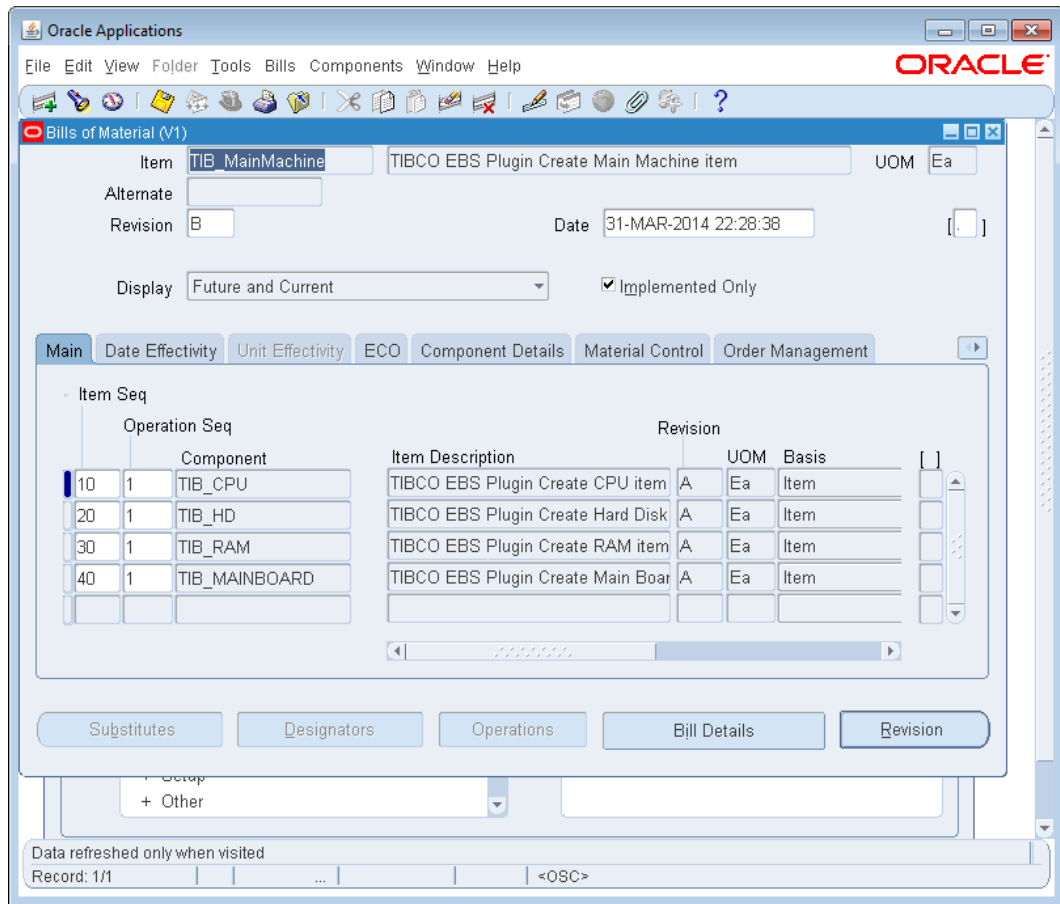
7. Click **Deselect All** in the **Applications** tab, and select the check box next to the project you want to run.
8. Click **Debug** to run the process.
9. Click the **Terminate**  icon in the Console view to stop the running process.  
You can check the value of **RETURN\_STATUS** in the **Output** tab in the Job Data view to see whether the process runs successfully. "s" stands for success, and "E" stands for error.
10. Request result in Oracle E-Business Suite:
  - a) Log in to the Oracle E-Business Suite system with the user name MFG and password welcome.
  - b) Click **Manufacturing and Distribution Manager > Bills of Materials > Bills > Bills** to open the Navigator - Manufacturing and Distribution Manager dialog.
  - c) Select v1 organization from the Organizations dialog, and click **OK**.



- d) Click the **Find**  icon.
- e) In the Find Bills (V1) dialog, specify the search filter in the **Item** field (for example, TIB\_%), and click **Find**.



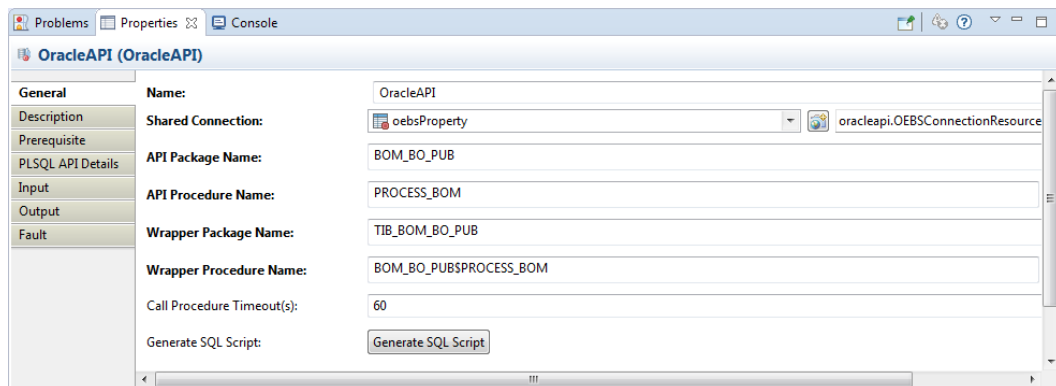
The following figure shows the result in the Oracle E-Business Suite web system:



## BOM\_BO\_PUB Package Configuration

You can use the BOM\_BO\_PUB package to create BOM bills.

The following figure shows a sample configuration of the BOM\_BO\_PUB package:



As defined in Oracle Integration Repository, you can use the BOM\_BO\_PUB.PROCESS\_BOM method for creating, updating, or deleting entities of a single Structure/BOM. The method takes in a single Structure/BOM header as well as all its components, revisions, reference designators, and substitute components.

The BOM\_BO\_PUB.PROCESS\_BOM (Process Single Structure/BOM) procedure has 23 parameters of PL/SQL Table types and PL/SQL Record types defined in the package. The following table lists the main parameters and associated inputs of BOM header, revision, components, reference designators, and substitute components:

	Parameter	Input
P_BOM_HEADER_REC	ASSEMBLY_ITEM_NAME	TIB_MainMachine
	ORGANIZATION_CODE	V1
	TRANSACTION_TYPE	CREATE
P_BOM_REVISION_TBL	REVISION	B
P_COMPONENT_TBL	COMPONENT_ITEM_NAME	TIB_CPU or TIB_HD or TIB_RAM or TIB_MAINBOARD
P_BOM_REF_DESIGNATOR_TBL	REF_DESIGNATOR_COMMENT	This is TIBCO CPU
P_BOM_SUB_COMPONENT_TBL	COMPONENT_ITEM_NAME	TIB_RAM
	SUBSTITUTE_COMPONENT_NAME	TIB_RAM1



You can define these parameters manually or automatically by defining a module property. For how to define module properties, see "Using Process and Module Properties" in *TIBCO ActiveMatrix BusinessWorks Samples*.

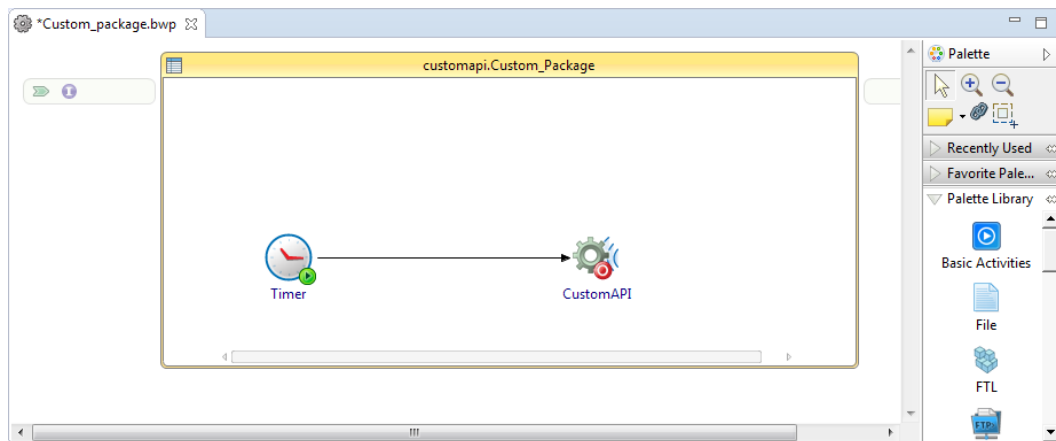
## Working with the CustomAPI Project

The CustomAPI project contains a process that demonstrates how to use the Custom API activity to call a PL/SQL procedure from Oracle Database Dictionary.



This example requires the use of the Vision Demo environment. It relies on responsibility, users, and data in the demo installation.

The process is designed with a Custom API activity as shown in the following figure:



Activity	Description
CustomAPI	Calls a PL/SQL procedure from Oracle Database Dictionary.



## Running the CustomAPI Project

You can run the CustomAPI project to see how to use the Custom API activity to call a PL/SQL procedure from Oracle Database Dictionary.



See [ACCOUNT\\_MGR Package Configuration](#) for details of the configuration and inputs of the activity.


### Prerequisites


Before running the project, ensure that you have connected to an Oracle Database Server and imported the project to TIBCO Business Studio. See [Connecting to Oracle Database Server](#) and [Importing Sample Projects](#) for more details.

### Procedure

1. In the Project Explorer view, expand the CustomAPI project.
2. Configure the Oracle E-Business Suite connection:
  - a) Expand **Resources** > **customapi**.
  - b) Double-click `OEBSConnectionResource.oebconnectionResource`.
  - c) In the OEBS Connection editor, edit the resource connection, and then click **Test Connection** to validate your connection.

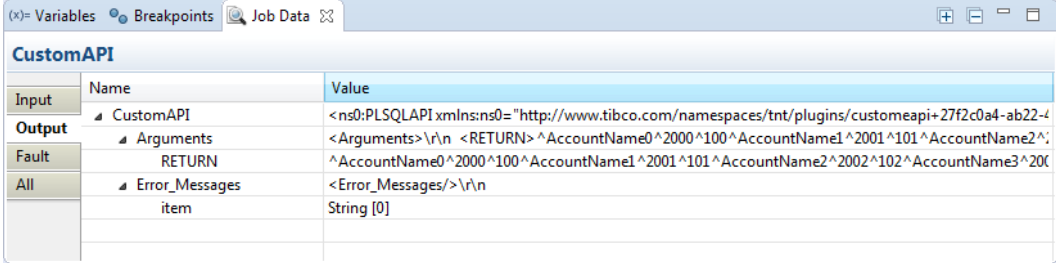
See [Oracle E-Business Suite Connection](#) for more details about how to configure the connection.

3. Open the process:
  - a) Expand **Processes** > **customapi**.
  - b) Double-click `Custom_package.bwp`.
4. Save the project.
5. From the menu, Click **Run** > **Debug Configurations**, or click  **Debug** > **Debug Configurations**.
6. Click **BusinessWorks Application** > **BWApplication** in the left panel in the Debug Configurations window.
 

By default, all applications in the current workspace are selected in the **Applications** tab.
7. Click **Deselect All** in the **Applications** tab, and select the check box next to the project you want to run.
8. Click **Debug** to run the process.
9. Click the **Terminate**  icon in the Console view to stop the running process.

You can check the value of RETURN in the **Output** tab in the Job Data view to see whether the process runs successfully.

The following figure shows a sample of the running result:

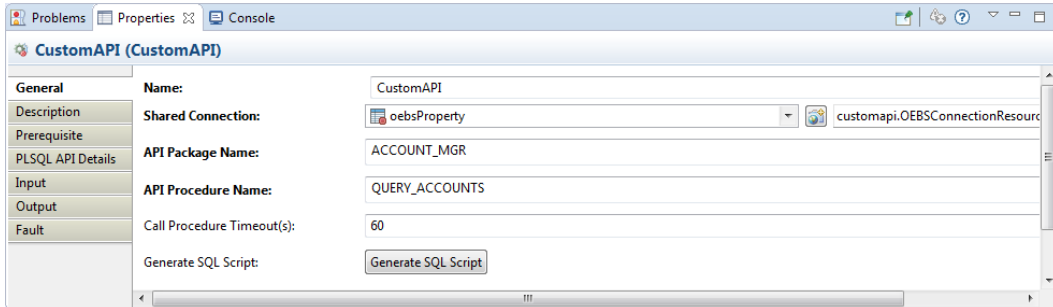


Input	Name	Value
Output	CustomAPI	<ns0:PLSQLAPI xmlns:ns0="http://www.tibco.com/namespaces/tnt/plugins/customapi+27f2c0a4-ab22-4
	Arguments	<Arguments>\r\n <RETURN> ^AccountName0^2000^100^AccountName1^2001^101^AccountName2^;
Fault	RETURN	^AccountName0^2000^100^AccountName1^2001^101^AccountName2^2002^102^AccountName3^200
All	Error_Messages	<Error_Messages/>\r\n
	item	String [0]

## ACCOUNT\_MGR Package Configuration

The ACCOUNT\_MGR package is the test package provided by Oracle. It is in Oracle Database Dictionary, not in Oracle Integration Repository.

The following figure shows a sample configuration of the ACCOUNT\_MGR package:



The following table lists the inputs of the ACCOUNT\_MGR package of the CustomAPI project:

Parameter	Type	Input
API_VERSION	NUMBER	1.0
P_PARTY_ID	NUMBER	4509

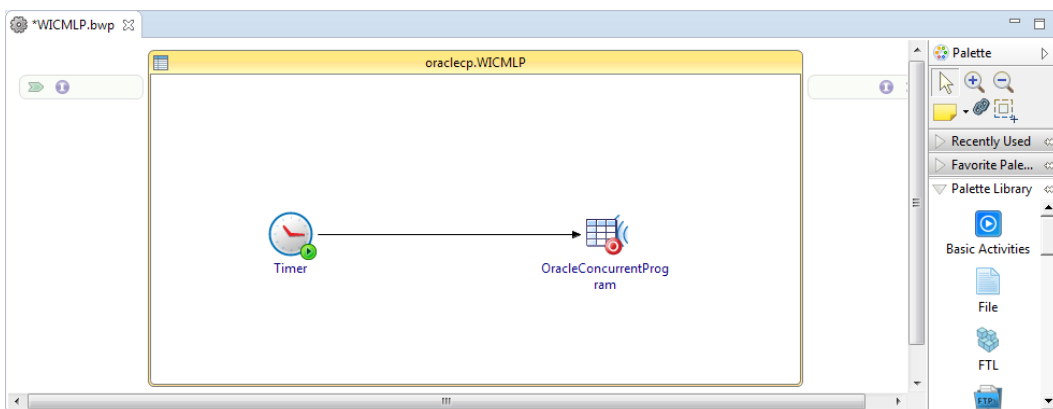
## Working with the OracleCP Project

The OracleCP project contains a process that demonstrates how to use the Oracle Concurrent Program WICMLP to call JDBC to insert data into interface tables and to submit a Concurrent Program request.



This example requires the use of the Vision Demo environment. It relies on responsibility, users, and data in the demo installation.

The process is designed with an Oracle CP activity as shown in the following figure:



Activity	Description
OracleConcurrentProgram	Calls JDBC to insert the jobs (tib_job01 and tib_job02) into the interface table WIP_JOB_SCHEDULE_INTERFACE and submits a Concurrent Program request.

## Running the OracleCP Project

You can run the OracleCP project to see how to use the Oracle Concurrent Program WICMLP to call JDBC to insert data into interface tables and to submit a Concurrent Program request.





See [WICMLP Concurrent Program Configuration](#) for details of the configuration and inputs of the activity.

### Prerequisites

Before running the project, ensure that you have connected to an Oracle Database Server and imported the project to TIBCO Business Studio. See [Connecting to Oracle Database Server](#) and [Importing Sample Projects](#) for more details.

### Procedure

1. In the Project Explorer view, expand the OracleCP project.
2. Configure the Oracle E-Business Suite connection:
  - a) Expand **Resources** > **oraclecp**.
  - b) Double-click `OEBSConnectionResource.oebconnectionResource`.
  - c) In the OEBS Connection editor, edit the resource connection, and then click **Test Connection** to validate your connection.

See [Oracle E-Business Suite Connection](#) for more details about how to configure the connection.
3. Open the process:
  - a) Expand **Processes** > **oraclecp**.
  - b) Double-click `WICMLP.bwp`.
4. Execute the generated SQL script:
  - a) Click **Generate SQL Script** in the **General** tab to generate the SQL scripts.  
See [Oracle Concurrent Program General Tab](#) for more details.
  - b) Execute the SQL script on the command line to grant insertion privileges to the plug-in user for the interface table.
5. Save the project.
6. From the menu, Click **Run** > **Debug Configurations**, or click  **Debug** > **Debug Configurations**.
7. Click **BusinessWorks Application** > **BWApplication** in the left panel in the Debug Configurations window.  
By default, all applications in the current workspace are selected in the **Applications** tab.
8. Click **Deselect All** in the **Applications** tab, and select the check box next to the project you want to run.
9. Click **Debug** to run the process.
10. Click the **Terminate**  icon in the Console view to stop the running process.  
You can check the value of **STATUS** in the **Output** tab in the Job Data view to see whether the process runs successfully. "Normal" stands for success, and "Error" stands for error.
11. Request result in Oracle E-Business Suite:
  - a) Log in to the Oracle E-Business Suite system with the user name MFG and password welcome.
  - b) Click **Manufacturing and Distribution Manager** > **Other** > **Concurrent** to open the Find Requests dialog.

Find Requests

My Completed Requests  
 My Requests In Progress  
 All My Requests  
 Specific Requests

Request ID: 5834307

Name: \_\_\_\_\_

Date Submitted: \_\_\_\_\_

Date Completed: \_\_\_\_\_

Status: \_\_\_\_\_

Phase: \_\_\_\_\_

Requestor: \_\_\_\_\_

Include Request Set Stages in Query

Order By: Request ID

Select the Number of Days to View: 7

Submit a New Request...    Clear    Find

c) Click **Find** to view the search results.

The following figure shows the search results:

Requests

Refresh Data    Find Requests    Submit a New Request...

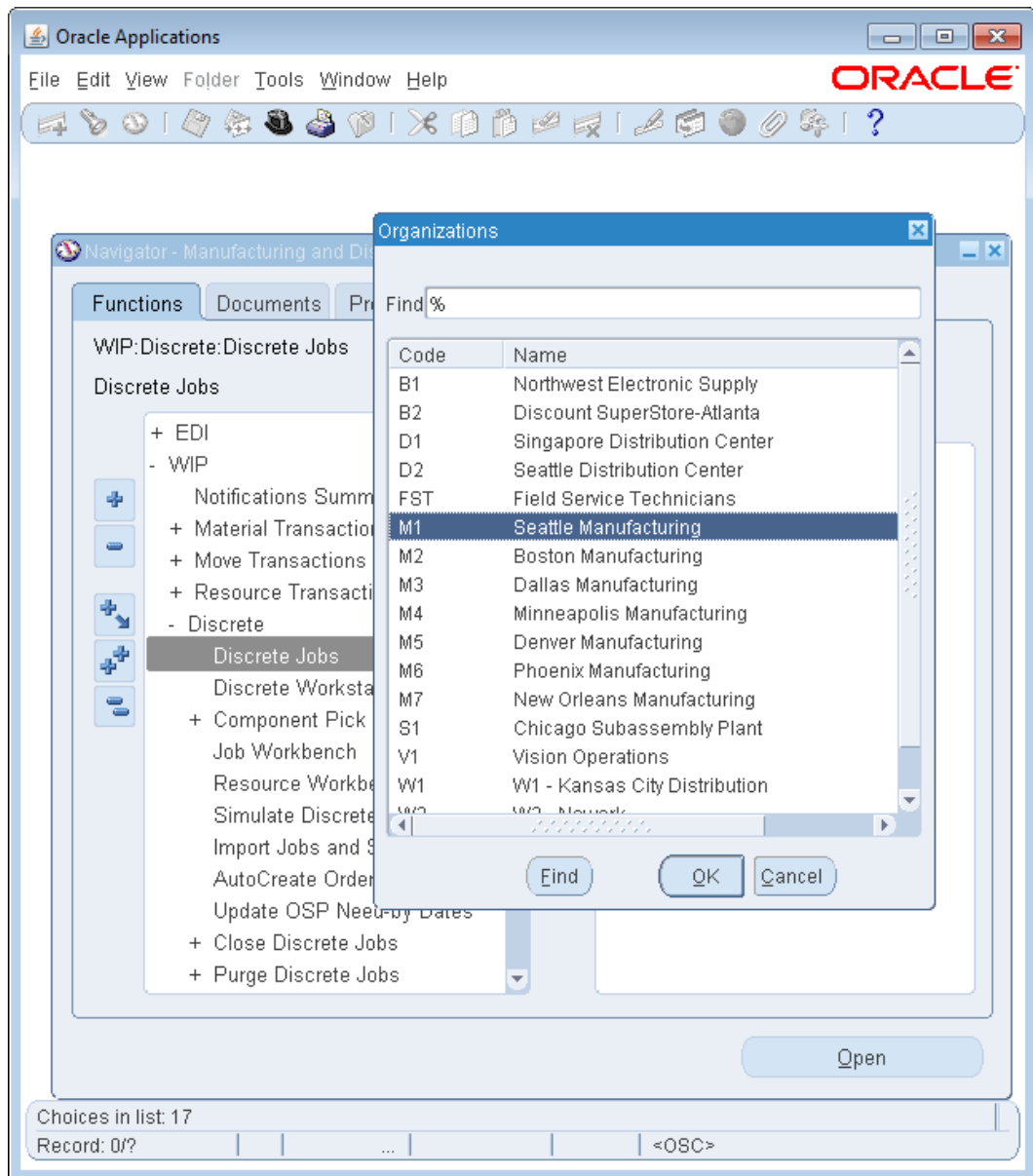
Request ID	Name	Parent	Phase	Status	Parameters
5834307	WIP Mass Load		Completed	Normal	201402181105, , 2

Hold Request    View Details...    View Output  
 Cancel Request    Diagnostics    View Log...

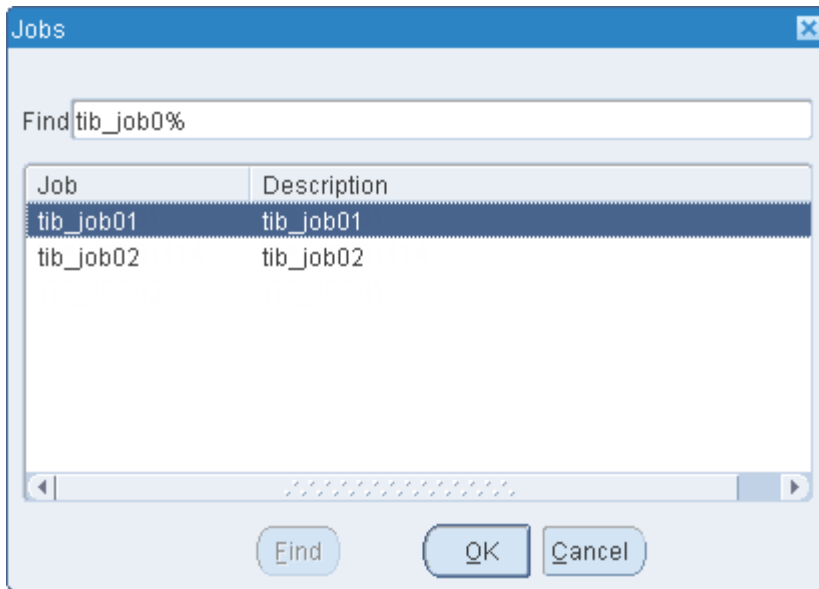
d) Close the Requests dialog.

e) In the Navigator - Manufacturing and Distribution Manager dialog, expand **WIP > Discrete**, and then double-click **Discrete Jobs**.

- f) In the Organizations dialog, select the M1 organization that is defined in the input, and then click **OK**.



- g) In the Find Discreet Jobs (M1) dialog, enter `tib_job0%` in the **Jobs** field, and then click **Find**. The following figure shows the Jobs dialog:



h) Click **OK** to view the Discrete Jobs details.

The following figure shows the Discrete Jobs details:

Job: tib\_job01      Type: Standard

Assembly: AT23808      Envoy Ambassador Laptop

Class: Discrete      UOM: Ea

Status: Unreleased       Firm

Quantities: Start: 12, MRP Net: 12

Dates: Start: 04-NOV-2010 10:10:00, Completion: 12-NOV-2010 12:35:00

Reference:      Revision: A      Revision Date: 04-NOV-2010 10:10:00

Supply Type: Based on Bill

Serial Numbers    Sales Orders    Operations    Components

### WICMLP Concurrent Program Configuration

The WICMLP Concurrent Program executes the open interface for work orders. You can use it to create a new job or schedule, or update an existing job or schedule information. It processes the records from the following tables: WIP\_JOB\_SCHEDULE\_INTERFACE and WIP\_JOB\_DTLS\_INTERFACE.

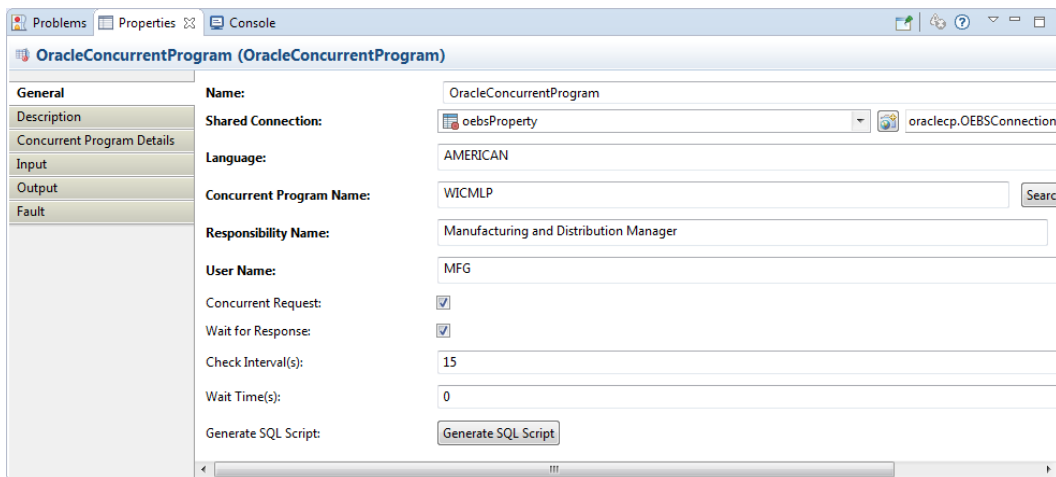
The WICMLP Concurrent Program includes the following parameters:

Name	Type	Required	Displayed	Description
<b>Group ID</b>	FND_NUMBER15_REQUIRED	Yes	Yes	Used to batch the interface records.
<b>Validation Level</b>	FND_NUMBER15	Yes	No	Indicates what validation should be performed.
<b>Print Report</b>	WIP_SRS_YES_NO_MAND	Yes	Yes	Indicates whether the report is printed.

The WICMLP Concurrent Program includes the following open interface tables or views:

Name	Direction	Status	Description
WIP_JOB_SCHEDULE_INTERFACE	Inbound	Active	Contains the requests to create or modify discrete jobs or repetitive schedules.
WIP_JOB_DTLS_INTERFACE	Inbound	Active	Contains requests to add, delete, and modify material and/or resource requirements for existing discrete jobs or repetitive schedules.

The following figure shows the WICMLP Concurrent Program configuration of the OracleCP project:



The responsibility and the user selected in this example are valid in the Vision Demo environment.

The following table lists the inputs of the WICMLP Concurrent Program of the OracleCP project:

Parameter	Type	Input
<b>Group ID</b>	VARCHAR2	See process property WIP_input/Request_GroupID
<b>Print Report</b>	VARCHAR2	2

The following table lists the important inputs of the interface tables of the OracleCP project:

Parameter	Type	Input
<b>Group ID</b>	NUMBER	See process property WIP_input/Request_GroupID
<b>SOURCE_CODE</b>	VARCHAR2	WICDOL
<b>ORGANIZATION_ID</b>	NUMBER	207
<b>CLASS_CODE</b>	VARCHAR2	Discrete
<b>JOB_NAME</b>	VARCHAR2	See process property WIP_input/JobName1
<b>DESCRIPTION</b>	VARCHAR2	See process property WIP_input/JobName1
<b>ORGANIZATION_CODE</b>	VARCHAR2	M1
<b>ROUTING_REVISION</b>	VARCHAR2	A
<b>BOM_REVISION</b>	VARCHAR2	A
<b>COMPLETION_SUBINVENTORY</b>	VARCHAR2	FGI
<b>ALLOW_EXPLOSION</b>	VARCHAR2	Y

If you use the Concurrent Program activity to insert data into the interface tables, you must clear the **Concurrent Request** check box in the **General** tab. You can also use the TIBCO BusinessWorks JDBC activity or other tools to insert records into the interface tables.



The **Group\_ID** input value must be identical within the example. Otherwise, an empty request is submitted.

## Working with the CustomCP Project

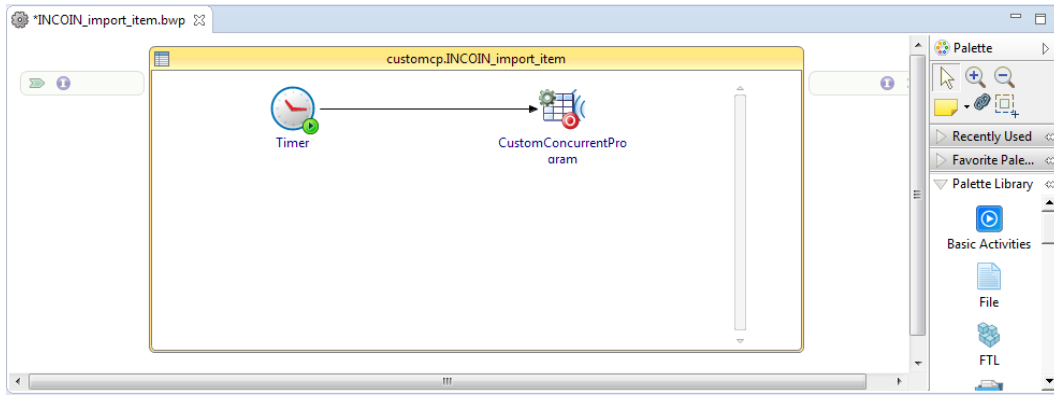
The CustomCP project contains a process that demonstrates how to use the custom Concurrent Program INCOIN to call JDBC to insert data into interface tables and to submit a Concurrent Program request.



This example requires the use of the Vision Demo environment. It relies on responsibility, users, and data in the demo installation.

The process is designed with a Custom CP activity as shown in the following figure:





Activity	Description
CustomConcurrentProgram	Calls JDBC to insert inventory items into the interface table INV.MTL_SYSTEMS_ITEM_INTERFACE and submits a Concurrent Program request.

## Running the CustomCP Project

You can run the CustomCP project to see how to use the custom Concurrent Program INCOIN to call JDBC to insert data into interface tables and to submit a Concurrent Program request.



See [INCOIN Concurrent Program Configuration](#) for details of the configuration and inputs of the activity.

### Prerequisites

Before running the project, ensure that you have connected to an Oracle Database Server and imported the project to TIBCO Business Studio. See [Connecting to Oracle Database Server](#) and [Importing Sample Projects](#) for more details.

### Procedure

1. In the Project Explorer view, expand the CustomCP project.
2. Configure the Oracle E-Business Suite connection:
  - a) Expand **Resources** > **customcp**.
  - b) Double-click `OEBSConnectionResource.oebconnectionResource`.
  - c) In the OEBS Connection editor, edit the resource connection, and then click **Test Connection** to validate your connection.

See [Oracle E-Business Suite Connection](#) for more details about how to configure the connection.

3. Open the process:
  - a) Expand **Processes** > **customcp**.
  - b) Double-click `INCOIN_import_item.bwp`.
4. Execute the generated SQL script:
  - a) Click **Generate SQL Script** in the **General** tab to generate the SQL scripts.  
See [Custom Concurrent Program General Tab](#) for more details.
  - b) Execute the SQL script on the command line to grant insertion privileges to the plug-in user for the interface table.
5. Execute the following SQL script to find the Concurrent Program parameters:

```
SELECT
```

```

cp.concurrent_program_name CP_Name, -- The Concurrent Program name
dfcu.end_user_column_name Column_name, -- The real argument name
lv.meaning data_type, -- The data type of argument
ffv.maximum_size, -- The length of the argument
dfcu.required_flag, -- The argument required or not
dfcu.display_flag, -- The argument displayed or not on Oracle Form
dfcu.default_value, -- The default value of the argument
dfcu.column_seq_num -- The argument sequence number
FROM fnd_concurrent_programs_vl cp
LEFT OUTER JOIN fnd_descr_flex_col_usage_vl dfcu
ON dfcu.descriptive_flexfield_name
   ='$SRS$. ' || cp.concurrent_program_name
LEFT OUTER JOIN fnd_flex_value_sets ffv
ON ffv.flex_value_set_id = dfcu.flex_value_set_id
LEFT OUTER JOIN fnd_lookup_values_vl lv
ON lv.lookup_code = ffv.format_type
AND lv.lookup_type = 'FIELD_TYPE'
AND lv.enabled_flag = 'Y'
AND lv.security_group_id = 0
AND lv.view_application_id = 0
WHERE cp.CONCURRENT_PROGRAM_NAME LIKE UPPER('&CONC_PROG_NAME' || '%')
ORDER BY cp.concurrent_program_name, dfcu.column_seq_num;

```

You can execute the SQL script in one of the following ways:


- Execute the SQL script on the command line under the apps user.  
If you fail to find any data by using this method, you must add the following SQL statements in front of the SQL script, and then run them together.  

```
ALTER SESSION SET
NLS_LANGUAGE='AMERICAN' ;
```
- Execute the SQL script with Oracle SQL Developer.  
If you fail to find any data by using this method, you must run the following SQL statements, and then run the SQL script.  


```
ALTER SESSION SET
NLS_LANGUAGE='AMERICAN' ;
```

The following table shows the results of the INCOIN Concurrent Program parameters:

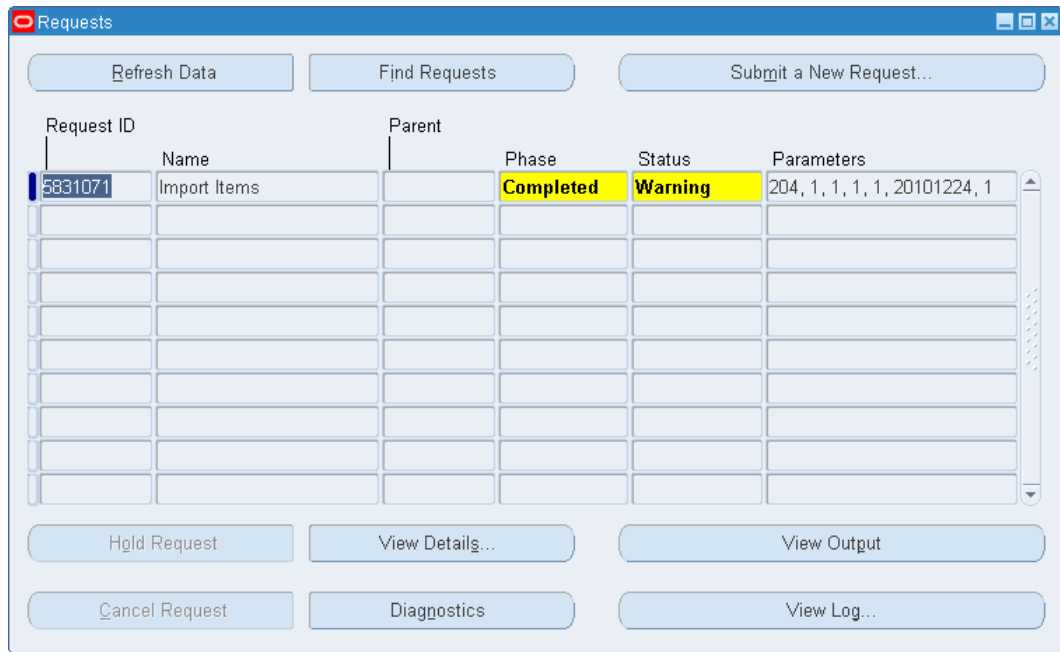
No	CP_NAME	ARGUMENT	COLUMN_NAME	DATA_TYPE	MAXIMUM_SIZE
1	INCOIN	argument1	p_org_id	Number	15
2	INCOIN	argument2	p_all_org	Char	80
3	INCOIN	argument3	p_val_item_flag	Char	80
4	INCOIN	argument4	p_pro_item_flag	Char	80
5	INCOIN	argument5	p_del_rec_flag	Char	80
6	INCOIN	argument6	p_xset_id	Number	15
7	INCOIN	argument7	p_run_mode	Char	1

- Save the project.
- From the menu, Click **Run > Debug Configurations**, or click  **Debug > Debug Configurations**.
- Click **BusinessWorks Application > BWApplication** in the left panel in the Debug Configurations window.

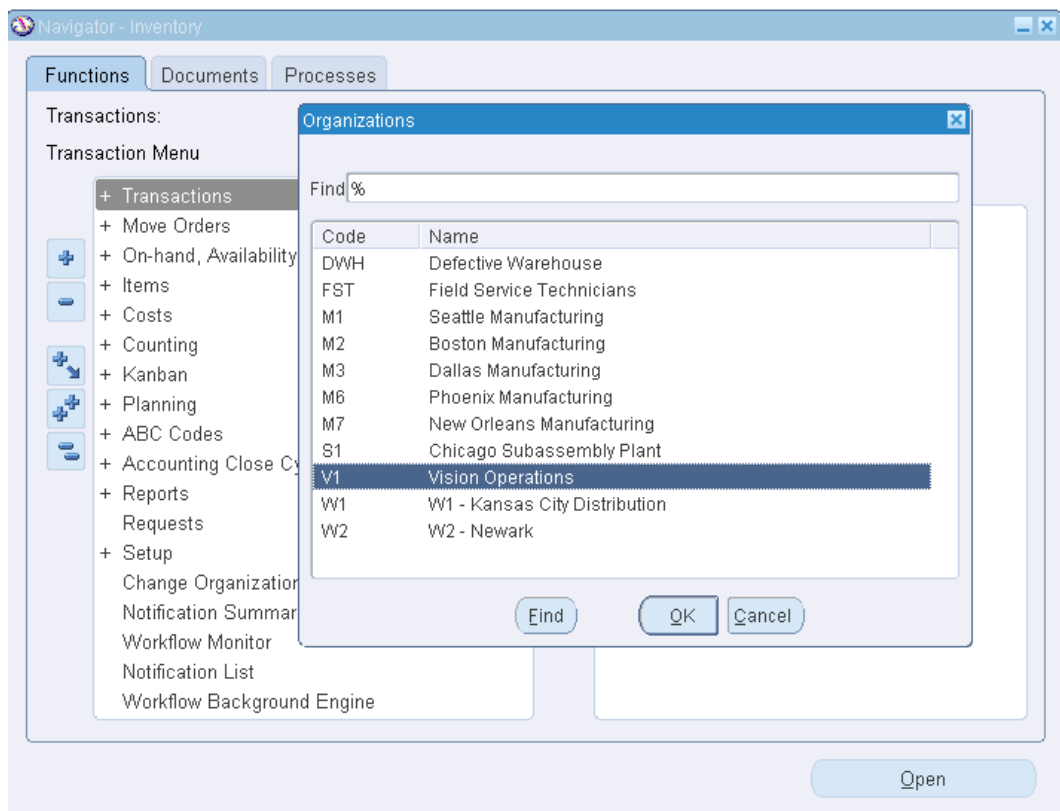
By default, all applications in the current workspace are selected in the **Applications** tab.

9. Click **Deselect All** in the **Applications** tab, and select the check box next to the project you want to run.
10. Click **Debug** to run the process.
11. Click the **Terminate**  icon in the Console view to stop the running process.  
You can check the value of **STATUS** in the **Output** tab in the Job Data view to see whether the process runs successfully. "Normal" stands for success, and "Error" stands for error.
12. Request result in Oracle E-Business Suite:
  - a) Log in to the Oracle E-Business Suite system with the user name MFG and password welcome.
  - b) Select **Manufacturing and Distribution Manager > Inventory > Requests** to open the Find Requests dialog.

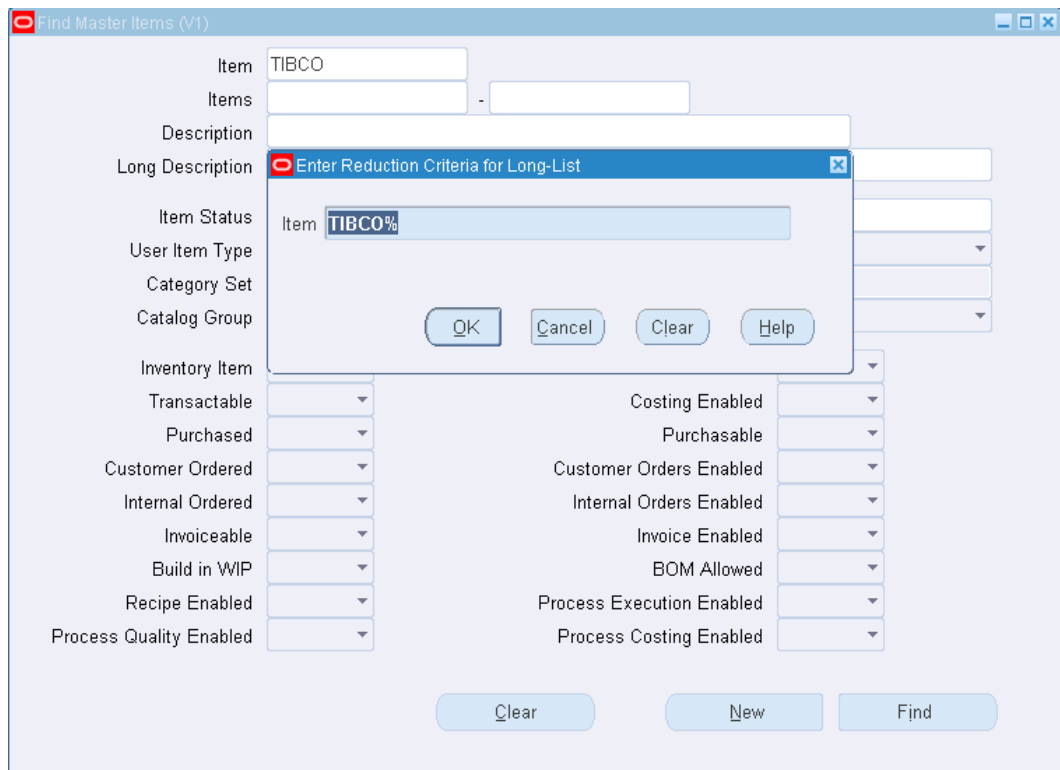
- c) Click **Find** to view the search results.  
The following figure shows the search results:



- d) Select **Inventory > Items > Master Items**.
- e) In the Organizations dialog, select the V1 organization that is defined in the input, and then click **OK**.

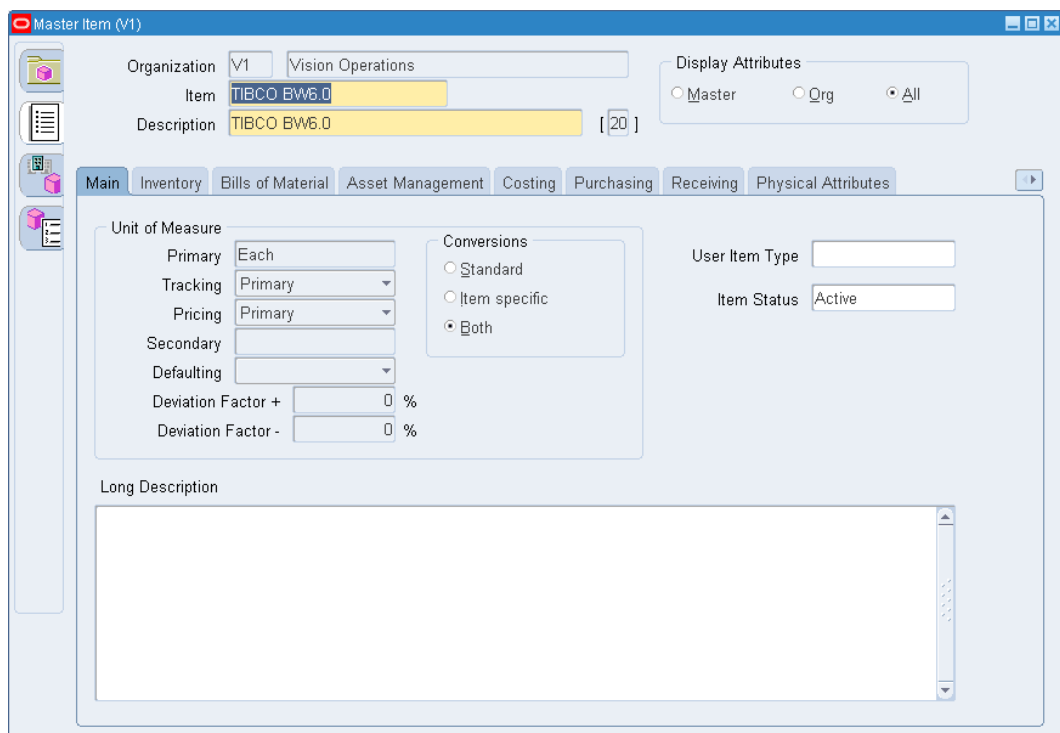


- f) In the **Item** field in the Find Master Items (V1) dialog, enter TIBCO%, and click **Find**.



g) In the Enter Reduction Criteria for Long-List dialog, click **OK** to view the search details.

The following figure shows the master item details:

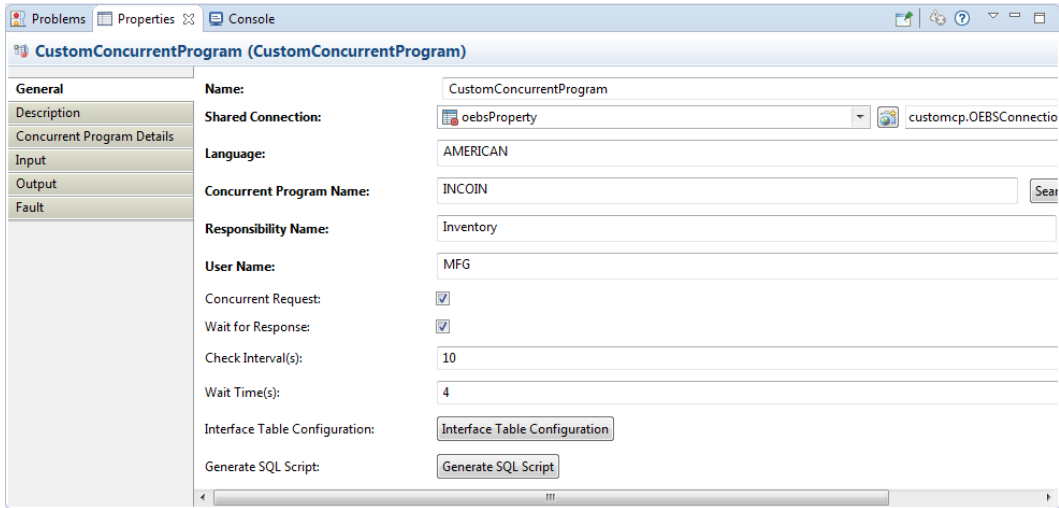


## INCOIN Concurrent Program Configuration

You can import items from any source into Oracle Inventory and Oracle Engineering by using the INCOIN Item Open Interface. With this interface, you can convert inventory items from another inventory system, migrate assembly and component items from a legacy manufacturing system,

convert purchased items from a custom purchasing system, and import new items from a product data management package.

The following figure shows a sample configuration of the INCOIN Concurrent Program:



## Parameters

The following table lists the inputs of the INCOIN Concurrent Program of the CustomCP project:

Parameter	Type	Input
argument1	VARCHAR2	204
argument2	VARCHAR2	1
argument3	VARCHAR2	1
argument4	VARCHAR2	1
argument5	VARCHAR2	1
argument6	VARCHAR2	See process property INCOIN_input/ arg6_set_process_id
argument7	VARCHAR2	1

## Open Interface Tables or Views

The INCOIN Concurrent Program contains 5 interface tables, as listed in the following table.

Name	Description
INV.MTL_SYSTEMS_ITEM_INTERFACE	Use this table for your new item numbers and all item attributes. This is the main Item Open Interface table. It might be the only table that you choose to use.

Name	Description
INV.MTL_ITEM_REVISIONS_INTERFACE	Use this table if you are importing revision details for your new items. This table is used only for revision information. It is not required.
INV.MTL_ITEM_CATEGORIES_INTERFACE	Use this table to import item category assignments. It stores data about item assignments to category sets and categories to be imported into the Oracle Inventory MTL_ITEM_CATEGORIES table.
INV.MTL_DESC_ELEM_VAL_INTERFACE	Use this table to describe elements that apply to your item.
INV.MTL_DESC_ELEM_VAL_INTERFACE	Use this table to record error messages for failed records in the interface table.



The open interface is not included in the Oracle E-Business Suite Integration Repository; therefore, you must go through the plug-in Custom Concurrent Program activity to access it. You must implement this activity by using the standard parameters for calling FND\_REQUEST.SUBMIT\_REQUEST; therefore, replace argument names with argument numbers for Concurrent Program.

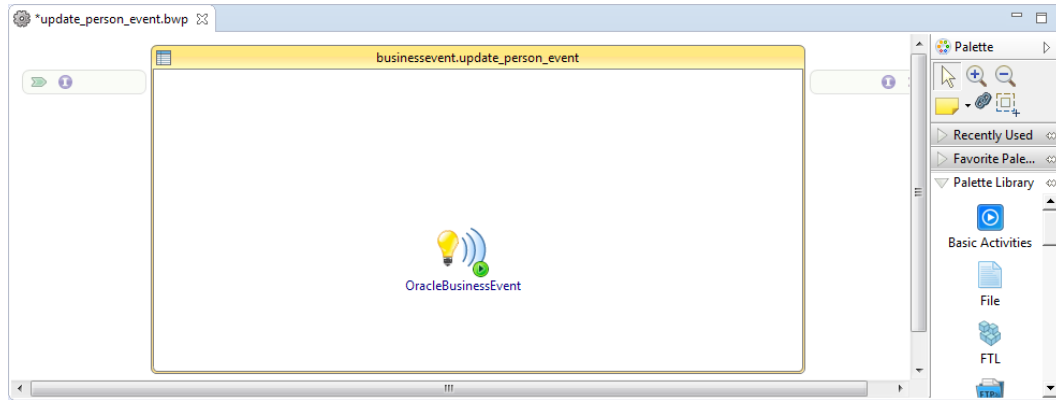
The following table lists the important inputs of the interface tables of the CustomCP project:

Parameter	Type	Input
ORGANIZATION_ID	NUMBER	204
DESCRIPTION	VARCHAR2	See process property INCOIN_input/ item_name
SEGMENT1	VARCHAR2	See process property INCOIN_input/ item_name
ATTRIBUTE1	VARCHAR2	2001
CUSTOMER_ORDER_FLAG	VARCHAR2	Y
CUSTOMER_ORDER_ENABLED_FLAG	VARCHAR2	Y
PROCESS_FLAG	NUMBER	1
ORGANIZATION_CODE	VARCHAR2	V1
TRANSACTION_TYPE	VARCHAR2	CREATE
SET_PROCESS_ID	NUMBER	See process property INCOIN_input/ arg6_set_process_id

## Working with the BusinessEvent Project

The BusinessEvent project contains a process that demonstrates how to use the Oracle Business Event oracle.apps.per.api.person.update\_person to listen to the update of a person's information from Oracle Advanced Queue.

The process is designed with an Oracle Business Event activity as shown in the following figure:



Activity	Description
OracleBusinessEvent	Listens to the update of a person's information from Oracle Advanced Queue.

## Running the BusinessEvent Project

You can run the BusinessEvent project to see how to use the Oracle Business Event oracle.apps.per.api.person.update\_person to listen to the update of a person's information from Oracle Advanced Queue.

### Prerequisites

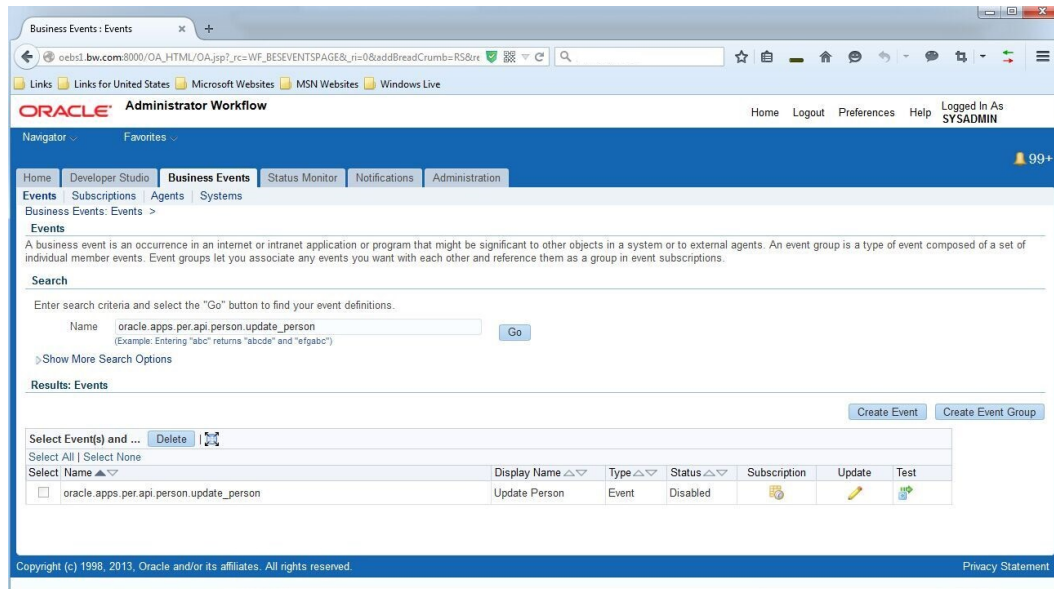
Before running the project, ensure that you have connected to an Oracle Database Server and imported the project to TIBCO Business Studio. See [Connecting to Oracle Database Server](#) and [Importing Sample Projects](#) for more details.

### Procedure


1. Review the business event:
  - a) Log in to the Oracle E-Business Suite website as the sysadmin user.
  - b) Select **Workflow Administrator Web Applications > Administrator Workflow > Business Events**.
  - c) Enter oracle.apps.per.api.person.update\_person in the **Name** field, and click **Go** to review the status of the business event.

The following figure shows the disabled business event:

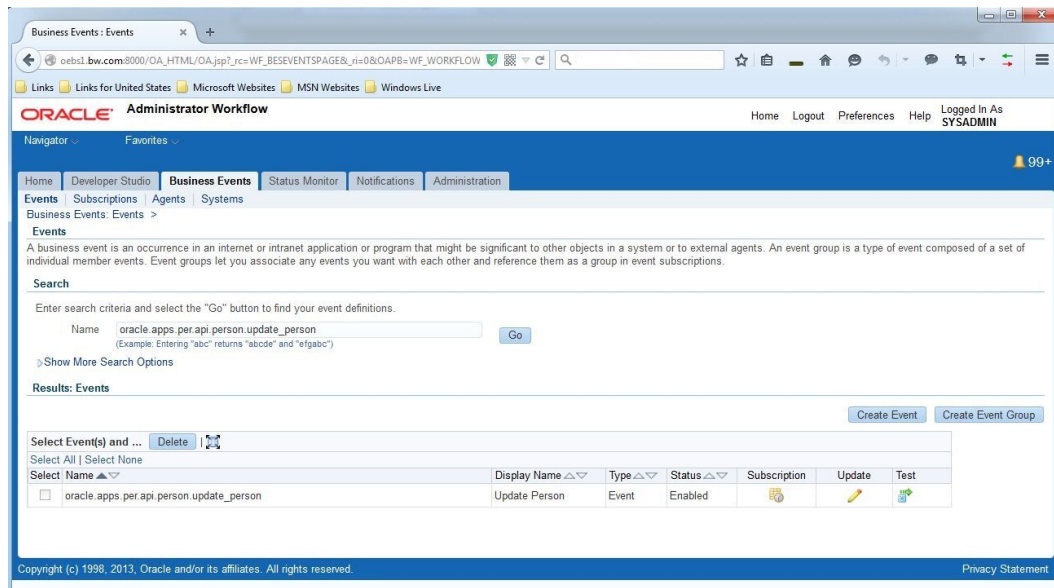




2. Enable the business event:

- a) Click the  icon to update the business event.
- b) In the **Status** list in the **Update Event** panel, select **Enabled** to enable the business events.
- c) Click **Apply**.


The following figure shows the enabled business event:



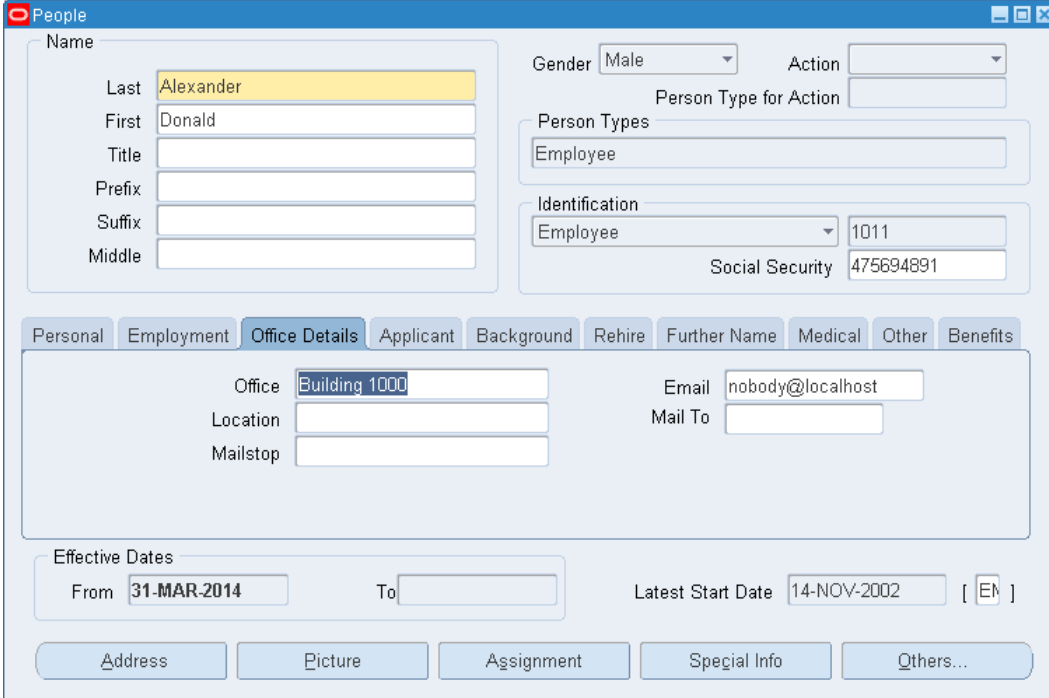
3. In the Project Explorer view, expand the BusinessEvent project.
4. Configure the Oracle E-Business Suite connection:
  - a) Expand **Resources > businesssevent**.
  - b) Double-click `OEBSConnectionResource.oebconnectionResource`.
  - c) In the OEBS Connection editor, edit the resource connection, and then click **Test Connection** to validate your connection.

See [Oracle E-Business Suite Connection](#) for more details about how to configure the connection.

5. Open the process:
  - a) Expand **Processes > businesssevent**.

- b) Double-click `update_person_event.bwp`.
6. Execute the generated SQL script:
  - a) Click **Generate SQL Script** in the **General** tab to generate the SQL scripts.  
See [Oracle Business Event General Tab](#) for more details.
  - b) Execute the SQL script on the command line to grant insertion privileges to the plug-in user for the interface table.
7. Trigger an event in Oracle E-Business Suite:
  - a) Log in to the Oracle E-Business Suite system as the Operations user. The default password is `welcome`.
  - b) Select **Human Resources, Vision Enterprises > People > Enter and Maintain**.
  - c) Enter `A%` in the **Full Name** field in the Find Person dialog to find a person's name that begins with an A, and then click **Find**.
  - d) Select `Alexander`, and click **OK** to review the person's information.
  - e) Update the value in the **Office** field from `Building 500` to `Building 1000`, and then click the  icon. Click **Update** in the pop-up dialog.

The following figure shows the updated personal information of Alexander:



The screenshot shows the 'People' form in Oracle E-Business Suite. The 'Name' section includes fields for Last (Alexander), First (Donald), Title, Prefix, Suffix, and Middle. The 'Gender' is set to 'Male'. The 'Person Types' section shows 'Employee' selected. The 'Identification' section shows 'Employee' selected, '1011' in the adjacent field, and 'Social Security' set to '475694891'. The 'Office Details' tab is active, showing 'Office' set to 'Building 1000', 'Location', and 'Mailstop' fields. The 'Email' field is 'nobody@localhost' and 'Mail To' is empty. The 'Effective Dates' section shows 'From' as '31-MAR-2014' and 'To' as empty. The 'Latest Start Date' is '14-NOV-2002'. At the bottom, there are buttons for 'Address', 'Picture', 'Assignment', 'Special Info', and 'Others...'.

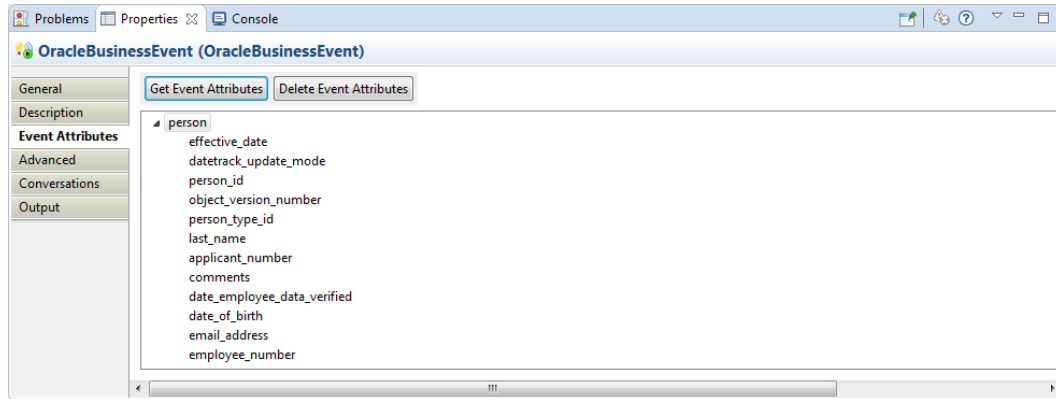
8. Get the event attributes in TIBCO Business Studio.



Click **Get Event Attributes** in the **Event Attributes** tab.

If you do not generate or execute the SQL script, an error `JAM-143: Queue must be specified` occurs.

If you do not trigger an event first, an error `Please trigger an event in Oracle E-Business Suite first` occurs.

The following figure shows the event attributes:



9. Save the project.
10. From the menu, Click **Run > Debug Configurations**, or click  **Debug > Debug Configurations**.
11. Click **BusinessWorks Application > BApplication** in the left panel in the Debug Configurations window.  
By default, all applications in the current workspace are selected in the **Applications** tab.
12. Click **Deselect All** in the **Applications** tab, and select the check box next to the project you want to run.
13. Click **Debug** to run the process.
14. Click the **Terminate**  icon in the Console view to stop the running process.

You can check the activity output in the **Output** tab in the Job Data view. The following figure shows a sample of the activity output:

Input	Name	Value
Output	last_medical_test_date	
	mailstop	
Fault	office_number	building 1000
All	on_military_service	N
	pre_name_adjunct	
	projected_start_date	
	rehire_authorizer	
	rehire_recommendation	N
	resume_exists	N
	resume_last_updated	
	second_passport_exists	N
	student_status	
	work_schedule	
	rehire_reason	

If you want to use the

`bw.application.job.flowlimit.application_name.application` VM argument when running the activity, you must set the `isFlowlimitForOebs` argument to true. The configuration of the argument is as follows:

- Design time: in the Run Configurations or Debug Configurations dialog, select **BusinessWorks Application > BApplication** on the left, and click the **Arguments** tab on the right; then enter `-DisFlowlimitForOebs=true` in the **VM arguments** text box.
- Run time: add `isFlowlimitForOebs=true` to the `config.ini` configuration file, which is located in the `TIBCO_HOME\bw\version_number\domains\domain_name\appnodes\appspace_name\appnode_name` directory.



# Managing Logs

When an error occurs, you can check logs to trace and troubleshoot plug-in exceptions.

By default, error logs are displayed in the Console view when you run a process in debug mode. You can change the log level of the plug-in to trace different messages and export logs to a file. Different log levels correspond to different messages, as described in [Log Levels](#).

## Log Levels

Different log levels include different information.

The plug-in supports the following log levels:

Log Level	Description
Debug	Indicates a developer-defined tracing message.
Info	Indicates normal plug-in operations. No action is required. A tracing message tagged with Info indicates that a significant processing step is reached, and logged for tracking or auditing purposes. Only info messages preceding a tracking identifier are considered as significant steps.
Warn	Indicates that an abnormal condition occurred. Processing continues, but for best practice, you can contact the administrator to investigate it.
Error	Indicates that an unrecoverable error occurred. Depending on the severity of the error, the plug-in might continue with the next operation or might stop.

## Setting up Log Levels

You can configure a different log level for the plug-in and plug-in activities to trace different messages.

By default, the plug-in uses the default log level of TIBCO ActiveMatrix BusinessWorks. The default log level of TIBCO ActiveMatrix BusinessWorks is **Error**.

### Procedure

1. Navigate to the `TIBCO_HOME\bw\version_number\config\design\logback` directory, and open the `logback.xml` file.
2. Add the following node in the **BusinessWorks Palette and Activity loggers** area to specify a log level for the plug-in:

```
<logger name="com.tibco.bw.palette.oeps.runtime">
  <level value="DEBUG"/>
</logger>
```

The value of the `level` element can be **Error**, **Info**, or **Debug**.



If you set the log level to **Debug**, the input and output for the plug-in activities are also displayed in the Console view. See [Log Levels](#) for more details regarding each log level.

3. Optional: add the following node in the **BusinessWorks Palette and Activity loggers** area to specify a log level for an activity:

```
<logger name="com.tibco.bw.palette.oeps.runtime.ActivityNameActivity">
  <level value="DEBUG"/>
</logger>
```

For example, add the following node to set the log level of the OracleAPI activity to Debug:

```
<logger name="com.tibco.bw.palette.oeps.runtime.OracleAPIActivity">
  <level value="DEBUG"/>
</logger>
```

However, for Oracle Business Event activities, the node added is somewhat different. For example:

```
<logger name="com.tibco.bw.palette.oeps.runtime.OracleBusinessEventEventSource">
  <level value="DEBUG"/>
</logger>
```



The activities that are not configured with specific log levels use the default log level of the plug-in.

4. Save the file.

## Exporting Logs to a File

You can update the `logback.xml` file to export plug-in logs to a file.

### Procedure

1. Navigate to the `TIBCO_HOME\bw\version_number\config\design\logback` directory, and open the `logback.xml` file.



After deploying an application in TIBCO Enterprise Administrator, navigate to the `TIBCO_HOME\bw\version_number\domains\domain_name\appnodes\appspace_name\appnode_name` directory to find the `logback.xml` file.

2. Add the following node to specify the file to which the log is exported:

```
<appender name="FILE" class="ch.qos.logback.core.FileAppender">
  <file>c:/bw6-oracleebs.log</file>
  <encoder>
    <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36}-%msg%n</pattern>
  </encoder>
</appender>
```

The value of the `file` element is the absolute path of the file that stores the exported logs.

3. Modify the root node at the bottom of the `logback.xml` file:

```
<root level="DEBUG">
  <appender-ref ref="STDOUT" />
  <appender-ref ref="FILE" />
</root>
```

4. Save the file.

# Updating the Oracle JDBC Library or Oracle Advanced Queuing Library

---

To update the Oracle JDBC Library or Oracle Advanced Queuing Library files, you must manually replace the .jar library files in the installation directories of the plug-in.

## Procedure

1. Exit TIBCO Business Studio.
2. Copy the .jar library files to both of the following directories:
  - `TIBCO_HOME\bw\palettes\oracleebs\version_number\design\plugins`
  - `TIBCO_HOME\bw\palettes\oracleebs\version_number\runtime\oracleebs.thirdparty.runtime\runtime\plugins\com.tibco.bw.palette.oracleebs.thirdparty`
3. Rename the files in both directories as follows:
  - Oracle JDBC library: `ojdbc.jar`
  - Oracle Advanced Queuing library: `aqapi.jar`

## Backing Up TIB\_BW\_EBS\_LOG Table

If you want to upgrade Oracle E-Business Suite 12.1.x to 12.2.x, you must back up the data in the TIB\_BW\_EBS\_LOG table.

Oracle E-Business Suite keeps a record of the plug-in activities in the TIB\_BW\_EBS\_LOG table. See [Log Table Structure](#) for details of the TIB\_BW\_EBS\_LOG table.

To back up the TIB\_BW\_EBS\_LOG table, you must perform the following operations:

- [Exporting Data from Oracle E-Business Suite 12.1.x](#)
- [Importing Data into Oracle E-Business Suite 12.2.x](#)

### Log Table Structure

Oracle E-Business Suite keeps a record of the plug-in activities in the TIB\_BW\_EBS\_LOG table.

The following table shows the structure of the TIB\_BW\_EBS\_LOG table:

Column Names	Data Type	Null?	Default Value	Description
<b>ID</b>	NUMBER	No		An auto increment sequence ID.
<b>PROCESS_ID</b>	VARCHAR2 (20)	Yes		Sequence ID; created automatically for each process by TIBCO BusinessWorks.
<b>NAME</b>	VARCHAR2 (100)	No		The name of the API/ Wrapper procedure, Concurrent Program, or business event.
<b>TYPE</b>	VARCHAR2 (50)	No		The type of the activity configured in TIBCO Business Studio.
<b>REQUEST_ID</b>	VARCHAR2 (20)	Yes		The ID of the request to run a Concurrent Program in Oracle E-Business Suite.
<b>LOG_TIME</b>	DATE	Yes	SYSDATE	The time when the process is recorded.
<b>STATUS</b>	VARCHAR2 (50)	No		The status of the process: Success, Error, or Ignore.

Column Names	Data Type	Null?	Default Value	Description
<b>EVENT_QUEUE</b>	VARCHAR2 (50)	Yes		The name of the Queue that receives the message from the Event Agent.  This field applies to Oracle Business Event only.
<b>EVENT_AGENT</b>	VARCHAR2 (50)	Yes		The name of the Agent that receives the message from the Oracle Business Event System.  This field applies to Oracle Business Event only.
<b>EVENT_SUBSCRIBER</b>	VARCHAR2 (50)	Yes		The name of the Subscriber that consumes the message.  This field applies to Oracle Business Event only.
<b>EVENT_DATA</b>	APPS.WF_EVENT_T	Yes		The user data of the business event message.  For more information, refer to the APPS.WF_EVENT_T data type definition in Oracle Database.  This field applies to Oracle Business Event only.
<b>DESCRIPTION</b>	VARCHAR2 (4000)	Yes		The simple-format output result for Oracle API, Custom API, Oracle CP, and Custom CP.



The `common_all.sql` script contains the creation script for the log table. It also contains the common procedure for writing records into the log table.



## Exporting Data from Oracle E-Business Suite 12.1.x

Before upgrading Oracle E-Business Suite 12.1.x to 12.2.x, you must back up the data in the TIB\_BW\_EBS\_LOG table in Oracle E-Business Suite 12.1.x.

### Procedure

1. Connect to the database of Oracle E-Business Suite 12.1.x under the plug-in user.
2. Run the TIB\_BW\_EBS\_LOG\_BACKUP\_TABLE.sql file located in the *TIBCO\_HOME*\bw\palettes\oracleebs\version\_number\config\12.2\upgrade\backupLogTable directory to create the TIB\_BW\_EBS\_LOG\_TEMP and WF\_PARAMETER\_T\_TEMP tables.
3. Run the TIB\_BW\_EBS\_LOG\_BACKUP.prc file located in the *TIBCO\_HOME*\bw\palettes\oracleebs\version\_number\config\12.2\upgrade\backupLogTable directory to create a stored procedure.
4. Issue the stored procedure created in [Step 3](#) to copy the data from the TIB\_BW\_EBS\_LOG table to the TIB\_BW\_EBS\_LOG\_TEMP and WF\_PARAMETER\_T\_TEMP tables.
5. Use the **exp** command provided by Oracle to export the TIB\_BW\_EBS\_LOG\_TEMP and WF\_PARAMETER\_T\_TEMP tables to your local machine.

The following example shows how to use the **exp** command. Enter the command at the command prompt.

```
exp plugin/<password>@<Net Service Name>
file=<D:\TIB_BW_EBS_LOG.dmp>
tables=(TIB_BW_EBS_LOG_TEMP,WF_PARAMETER_T_TEMP)
```

In this example, **plugin** stands for the plug-in user name, **password** stands for the password of the plug-in user, and **Net Service Name** stands for the name of the net service that you use. After running the command, the TIB\_BW\_EBS\_LOG\_TEMP and WF\_PARAMETER\_T\_TEMP tables are exported to the D:\TIB\_BW\_EBS\_LOG.dmp directory of your local machine.



Before using the **exp** command, you must install the corresponding Oracle database client. For Oracle E-Business Suite 12.1.x, you must install the Oracle 11gR1 client.

6. Run the post\_export.sql file located in the *TIBCO\_HOME*\bw\palettes\oracleebs\version\_number\config\12.2\upgrade\backupLogTable directory to drop the TIB\_BW\_EBS\_LOG\_TEMP and WF\_PARAMETER\_T\_TEMP tables and the stored procedure created by the TIB\_BW\_EBS\_LOG\_BACKUP.prc file from the database of Oracle E-Business Suite 12.1.x.

## Importing Data into Oracle E-Business Suite 12.2.x

After upgrading Oracle E-Business Suite to 12.2.x, you can import the data, which have been exported from Oracle E-Business Suite 12.1.x, into Oracle E-Business Suite 12.2.x.

### Procedure

1. Connect to the database of Oracle E-Business Suite 12.2.x under the plug-in user.
2. Use the **imp** command provided by Oracle to import the TIB\_BW\_EBS\_LOG\_TEMP and WF\_PARAMETER\_T\_TEMP tables from your local machine into the database of Oracle E-Business Suite 12.2.x.

The following example shows how to use the **imp** command. Enter the command at the command prompt.

```
imp plugin/<password>@<Net Service Name>
file=<D:\TIB_BW_EBS_LOG.dmp>
tables=(TIB_BW_EBS_LOG_TEMP,WF_PARAMETER_T_TEMP) ignore=y
```

In this example, **plugin** stands for the plug-in user name, **password** stands for the password of the plug-in user, and **Net Service Name** stands for the name of the net service that you use. After running the command, the TIB\_BW\_EBS\_LOG\_TEMP and WF\_PARAMETER\_T\_TEMP tables are imported from the D:\TIB\_BW\_EBS\_LOG.dmp directory of your local machine into the database of Oracle E-Business Suite 12.2.x.



Before using the **imp** command, you must install the corresponding Oracle database client. For Oracle E-Business Suite 12.2.x, you must install the Oracle 11gR2 client.

3. Run the TIB\_BW\_EBS\_LOG\_IMPORT.prc file located in the *TIBCO\_HOME*\bw\palettes\oracle\ebs\version\_number\config\12.2\upgrade\backupLogTable directory to create a stored procedure.
4. Execute the stored procedure created in [Step 3](#) to copy the data from the TIB\_BW\_EBS\_LOG\_TEMP and WF\_PARAMETER\_T\_TEMP tables to the TIB\_BW\_EBS\_LOG table in the database of Oracle E-Business Suite 12.2.x.



Make sure that the TIB\_BW\_EBS\_LOG table is empty before importing the data.

5. Run the post\_import.sql file located in the *TIBCO\_HOME*\bw\palettes\oracle\ebs\version\_number\config\12.2\upgrade\backupLogTable directory to drop the TIB\_BW\_EBS\_LOG\_TEMP and WF\_PARAMETER\_T\_TEMP tables and the stored procedure created by the TIB\_BW\_EBS\_LOG\_IMPORT.prc file from the database of Oracle E-Business Suite 12.2.x.

## Oracle API and Custom API Comparison

---

In TIBCO Business Studio, Oracle API and Custom API use different activities.

- Oracle API loads the PL/SQL package and procedure parameters information from Oracle Integration Repository.
- Custom API loads the PL/SQL package or procedure parameters information from Oracle Database Dictionary.

# Oracle Concurrent Program and Custom Concurrent Program Comparison

---

The configurations and types of concurrent programs supported by Oracle Concurrent Program and Custom Concurrent Program are different.

The differences between Oracle Concurrent Program and Custom Concurrent Program are listed as follows:

- Configuration
  - In TIBCO Business Studio, Oracle Concurrent Program and Custom Concurrent Program use different palettes:
    - Oracle Concurrent Program loads the Inbound Interface Tables or Views from Oracle Integration Repository.
    - Custom Concurrent Program requires you to select Interface Tables or Views from the database manually.
- Concurrent Program

Oracle Integration Repository contains two types of Concurrent Programs:

- Standard Request Submission (SRS) Concurrent Program
- Non-SRS Concurrent Program

The concurrent program activities supported by each activity are different:

- Oracle Concurrent Program only supports SRS Concurrent Program.
- Custom Concurrent Program supports both SRS Concurrent Program and non-SRS Concurrent Program.

See [SRS Concurrent Program and Non-SRS Concurrent Program](#) for more details about the differences between the two types of Concurrent Programs and how to use them.

## SRS Concurrent Program and Non-SRS Concurrent Program

Depending on the approach used to submit a request in Oracle E-Business Suite, two types of Concurrent Programs are available in Oracle Integration Repository.

- *Standard Request Submission (SRS)* is an Oracle E-Business Suite feature with which you can select and run your Concurrent Programs from a single, standard form (Submit Request) or window (Schedule Request). Requests to run Concurrent Programs are called concurrent requests.
- *Non-Standard Request Submission* is not available through Standard Request Submission. You can run a non-SRS Concurrent Program by submitting a non-standard request form.

This section explains how to check the type and parameters of a Concurrent Program, and demonstrates the correct and incorrect activity configurations when invoking a sample non-SRS Concurrent Program, GLBBSU. If you are unfamiliar with the concepts, it is good practice to read through the following topics. If you already have some knowledge about these concepts, you can read only [Using the Custom Concurrent Program Activity](#) for the right configuration by using the plug-in.

- [Sample Non-SRS Concurrent Program: GLBBSU](#)
- [Checking the Type and Parameters of a Concurrent Program](#)
- [Submitting a Request by Using Oracle E-Business Suite Client](#)
- [Submitting a Request by Using Plug-in Activities](#)

## Sample Non-SRS Concurrent Program: GLBBSU

GLBBSU is a non-SRS Concurrent Program, which means you cannot run the program by submitting a standard request form provided by Oracle E-Business Suite. To run the program, you must submit a non-standard request form.

The following figure shows the details of GLBBSU:

The screenshot displays the Oracle Integration Repository interface. The main content area shows the details for the 'Open Interface : Upload Budget Amounts' program. The 'Parameters' section is highlighted with a red box and contains the following table:

Name	Type	Required	Displayed	Description
Access Set ID	GL_SRS_NULL_NUM	Yes	No	Access Set ID
Ledger ID	GL_SRS_NULL_NUM	Yes	No	Ledger ID

You can view the details of a Concurrent Program in Oracle Integration Repository available with Oracle E-Business Suite. See [Introduction to Oracle Integration Repository](#) for details.

## Checking the Type and Parameters of a Concurrent Program

Before you make decision on which activity to use, check the type of the Concurrent Program. In addition, you must validate the parameters before starting the activity.



Some parameters for non-SRS Concurrent Programs are not visible from both TIBCO Business Studio and Oracle Integration Repository. To run such non-SRS Concurrent Programs, it is good practice to use the Custom Concurrent Program activity.

## Checking the Type of a Concurrent Program

Check the type of a Concurrent Program before you make decision on which activity to use. For an SRS Concurrent Program, you can use either the Oracle Concurrent Program activity or the Custom

Concurrent Program activity. For a non-SRS Concurrent Program, you can only use the Custom Concurrent Program activity.

### Procedure

1. Log on to Oracle SQL\*Plus.  
On the command line, enter `sqlplus`. Enter the Oracle Database system administrator's user name and password.
2. Run the SQL query for the Concurrent Program of your interest.  
For example:  

```
select CONCURRENT_PROGRAM_ID, CONCURRENT_PROGRAM_NAME,
SRS_FLAG,ENABLED_FLAG,USER_CONCURRENT_PROGRAM_NAME from
fnd_concurrent_programs_vl where concurrent_program_name='GLBBSU'
```

Y stands for SRS Concurrent Program, and N stands for non-SRS Concurrent Program.

### Finding the Parameters for a Concurrent Program

You must validate the parameters of the Concurrent Program before starting the activity.

### Procedure

1. Log on to Oracle SQL\*Plus.  
On the command line, enter `sqlplus` . Enter the Oracle Database system administrator's user name and password.
2. Run the SQL query for the Concurrent Program of your interest.  
For example:  

```
SELECT
cp.concurrent_program_name CP_Name, -- The Concurrent Program name
dfcu.end_user_column_name Column_name, -- The real argument name
lv.meaning data_type, -- The data type of argument
ffv.maximum_size, -- The length of the argument
dfcu.required_flag, -- The argument required or not
dfcu.display_flag, -- The argument displayed or not on Oracle Form
dfcu.default_value, -- The default value of the argument
dfcu.column_seq_num -- The argument sequence number
FROM fnd_concurrent_programs_vl cp
LEFT OUTER JOIN fnd_descr_flex_col_usage_vl dfcu
ON dfcu.descriptive_flexfield_name
= '$SRS$. ' || cp.concurrent_program_name
LEFT OUTER JOIN fnd_flex_value_sets ffv
ON ffv.flex_value_set_id = dfcu.flex_value_set_id
LEFT OUTER JOIN fnd_lookup_values_vl lv
ON lv.lookup_code = ffv.format_type
AND lv.lookup_type = 'FIELD_TYPE'
AND lv.enabled_flag = 'Y'
AND lv.security_group_id = 0
AND lv.view_application_id = 0
WHERE cp.CONCURRENT_PROGRAM_NAME = 'GLBBSU'
ORDER BY cp.concurrent_program_name, dfcu.column_seq_num;
```

## Submitting a Request by Using Oracle E-Business Suite Client

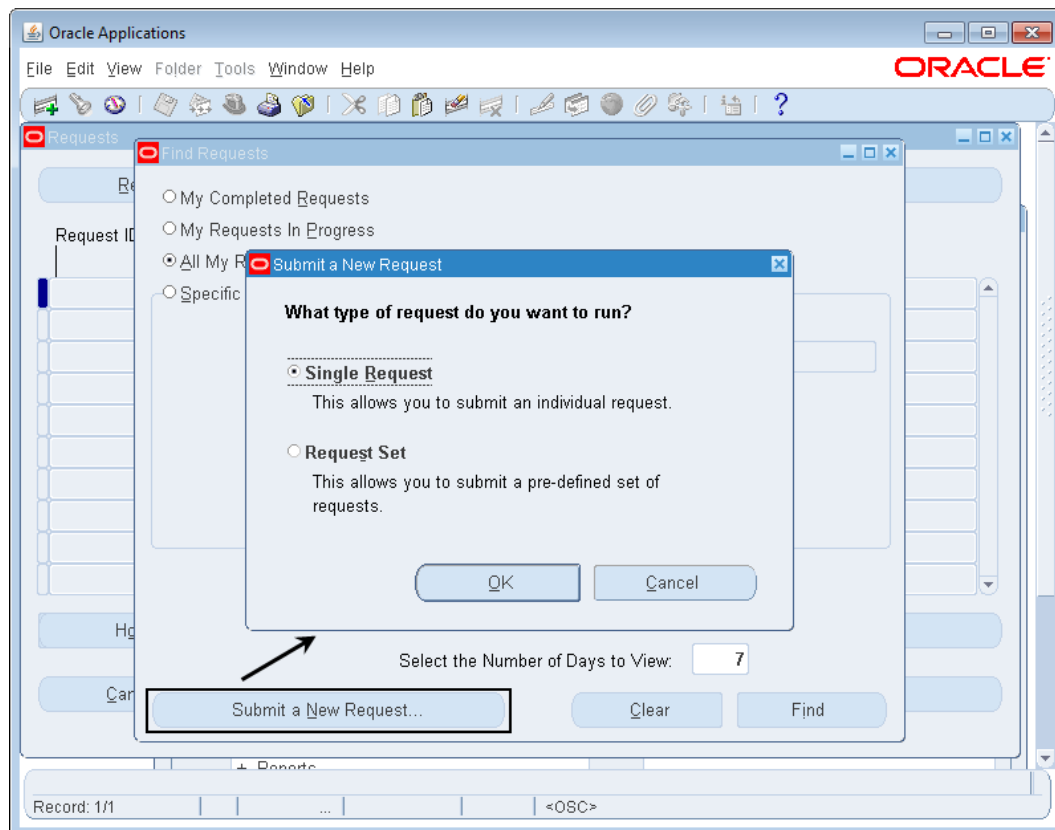
You can run a non-SRS Concurrent Program by submitting a non-standard request form. Non-Standard Request Submission is not available through Standard Request Submission.

### Submitting a Standard Request Form

Because non-Standard Request Submission is not available through Standard Request Submission, you cannot find the GLBBSU Concurrent Program when submitting a standard request form for it.

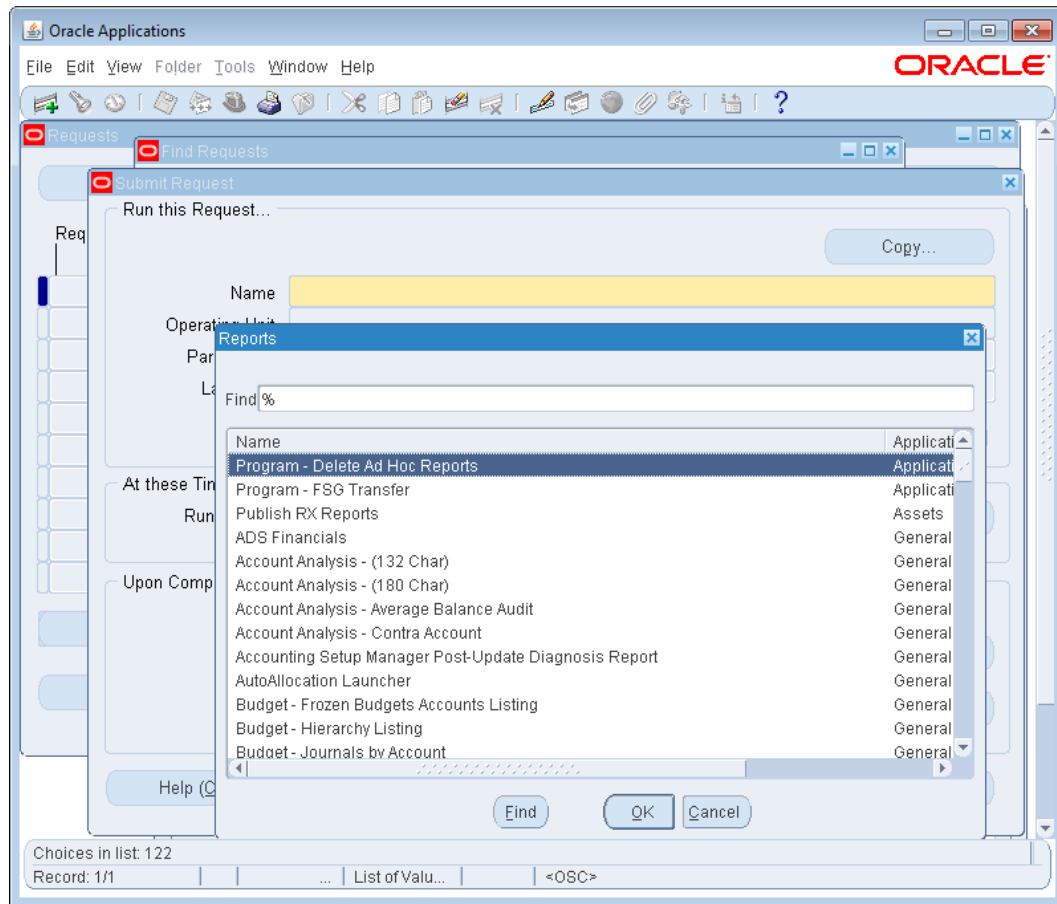
#### Procedure

1. Log on to Oracle E-Business Suite Client as the Operations user.
2. Select **General Ledger, Vision Operations (USA) > Other > Requests**.
3. In Oracle Applications, click **Submit a New Request** in the Find Requests dialog.



4. In the Submit a New Request dialog, click **Single Request**, and click **OK**.
5. In the Submit Request dialog, click the browse button next to the **Name** field.

The Reports dialog is displayed as shown in the following figure. But because GLBBSU (Upload Budget Amounts) is a non-SRS Concurrent Program, you cannot find it in the listed programs.



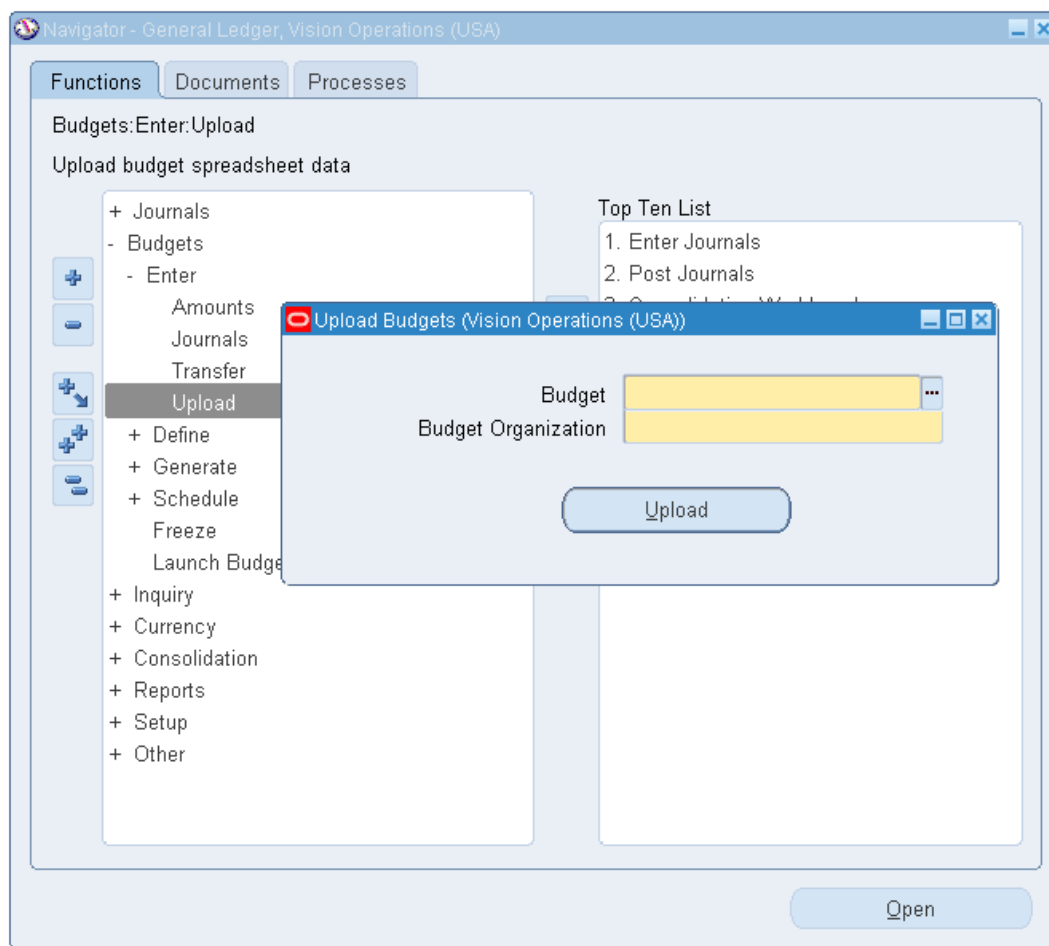
## Submitting a Non-Standard Request Form

You can run the GLBBSU Concurrent Program by submitting a non-standard request form.

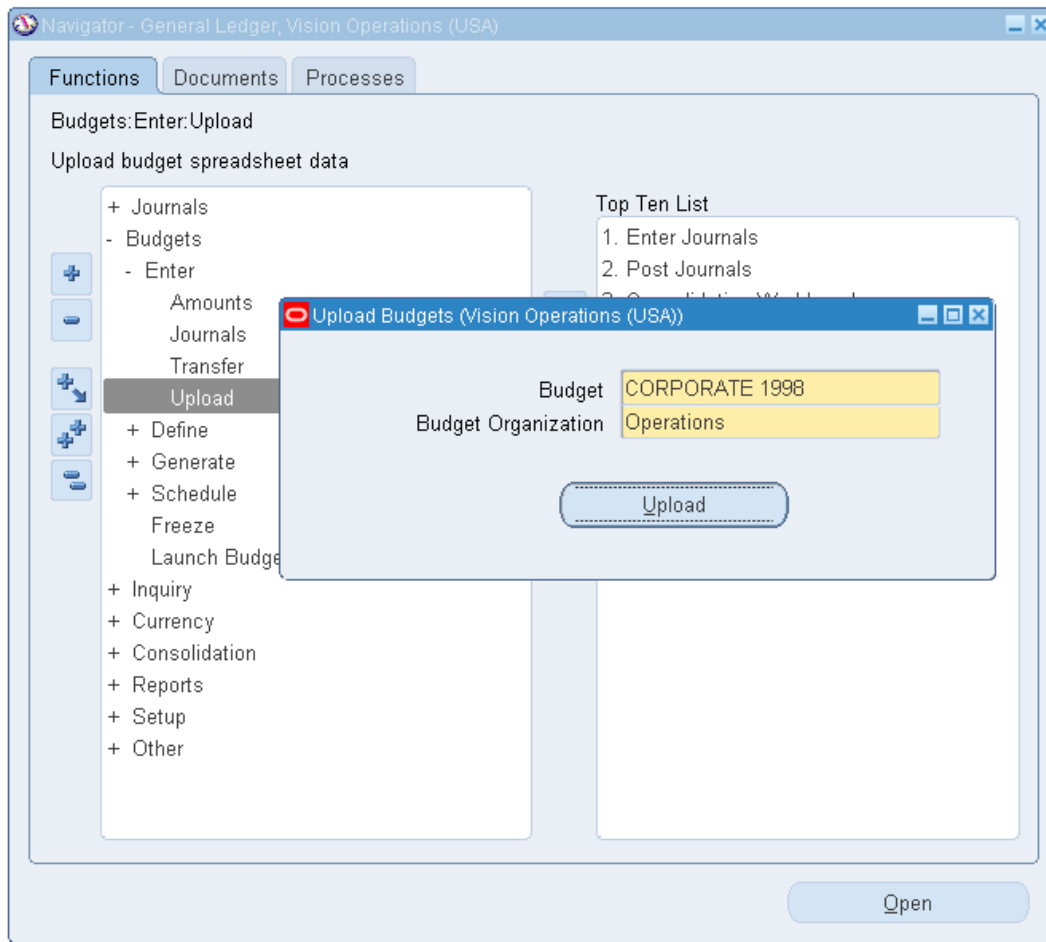
### Procedure

1. Log on to Oracle E-Business Suite Client as the Operations user.
2. Select **General Ledger, Vision Operations (USA) > Other > Requests**.
3. In Oracle Applications, close the Find Requests dialog.
4. In the Navigator dialog, expand **Budgets > Enter > Upload**, and double-click **Upload** to display the Upload Budgets dialog.





5. Enter the inputs for this Concurrent Program.  
The following figure shows a sample of the inputs:



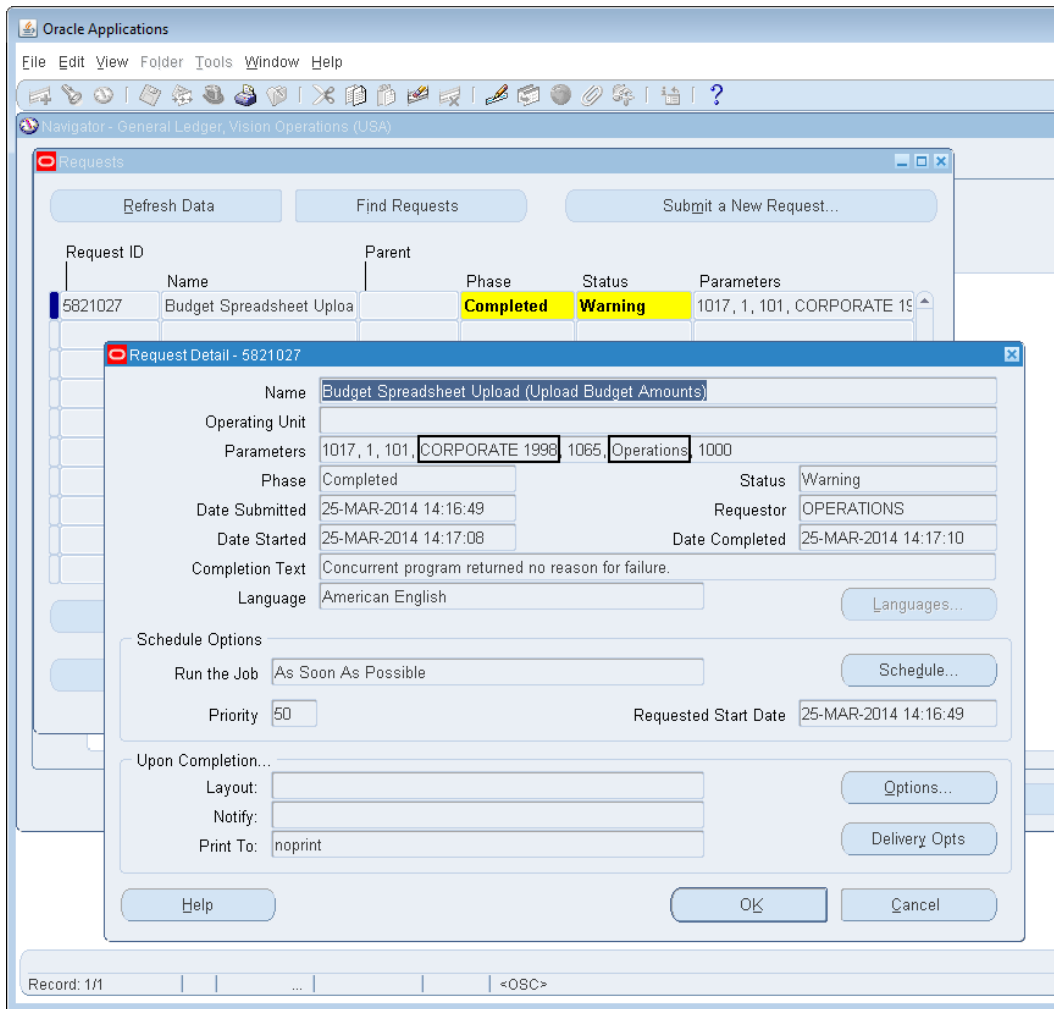
The inputs are different from the parameters shown in Oracle Integration Repository.

6. Click **Upload** to submit the request.

### Result

In Oracle Applications, click **View > Requests** from the main menu. In the Find Requests dialog, specify the search criteria and click **Find**.

The following figure shows the request details in Oracle E-Business Suite Client:



## Submitting a Request by Using Plug-in Activities

You must use the Custom Concurrent Program activity when running a non-SRS Concurrent Program. If you use the Oracle Concurrent Program activity, it cannot execute the request successfully and results in an error.

## Using the Oracle Concurrent Program Activity

When you run the GLBBSU Concurrent Program in the Oracle Concurrent Program activity, the request is executed with an error .

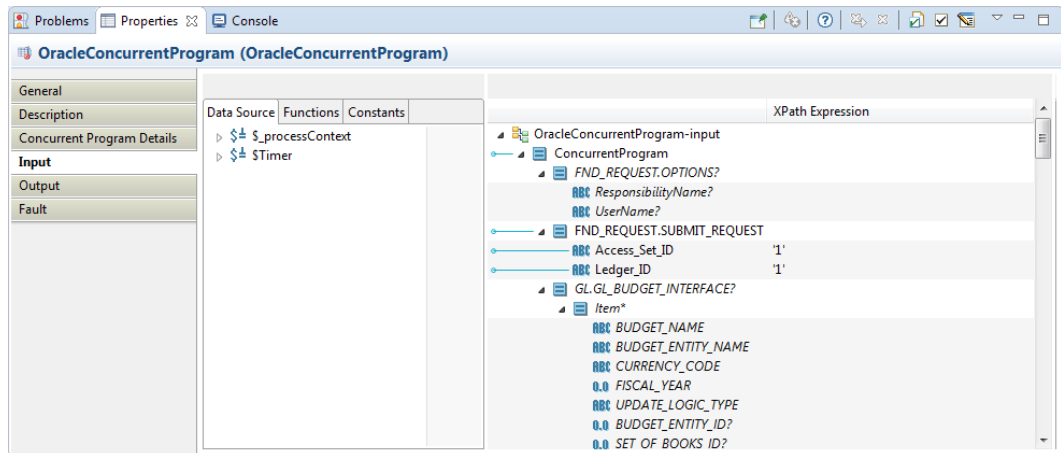
### Prerequisites

1. Create a project in TIBCO Business Studio.
2. Create a process in the project.

### Procedure

1. Add the Oracle Concurrent Program activity to the process.
2. Configure the activity and its inputs.

The following figure shows a sample configuration of the inputs:



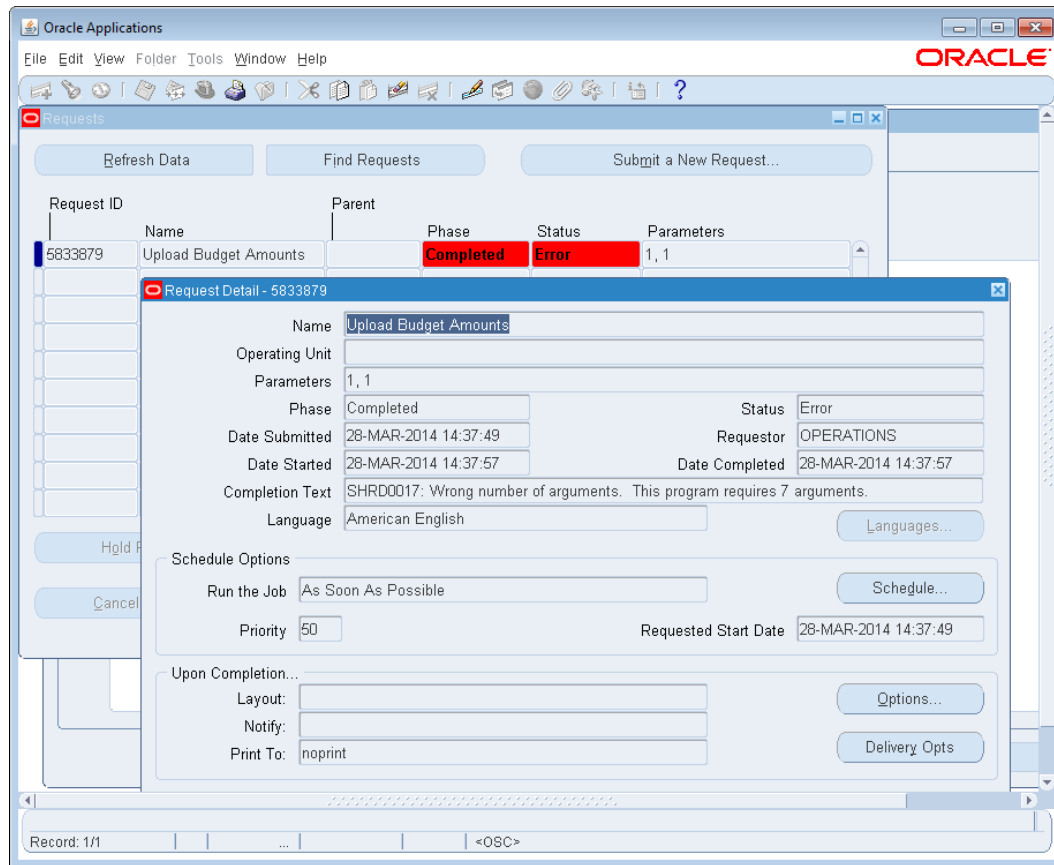
3. Run the process in Debug mode.

## Result

- Output in TIBCO Business Studio: the following figure shows the output of the activity. The **STATUS** field is **Error**, indicating that the process runs with an error, and the request is executed with an error.

OracleConcurrentProgram		
Input	Name	Value
Output	OracleConcurrentProgram	<ns0:ConcurrentProgram xmlns:ns0="http://www.tibco.com/namespaces/tnt/plugins/oracleconcurrentp
	REQUEST_ID	5832316
Fault	PHASE	Completed
	STATUS	Error
All	MESSAGE	SHRD0017: Wrong number of arguments. This program requires 7 arguments.\n

- Request results in Oracle E-Business Suite: the following figure shows the request details in Oracle E-Business Suite.



## Using the Custom Concurrent Program Activity

You can run the GLBBSU Concurrent Program successfully in the Custom Concurrent Program activity.

### Prerequisites

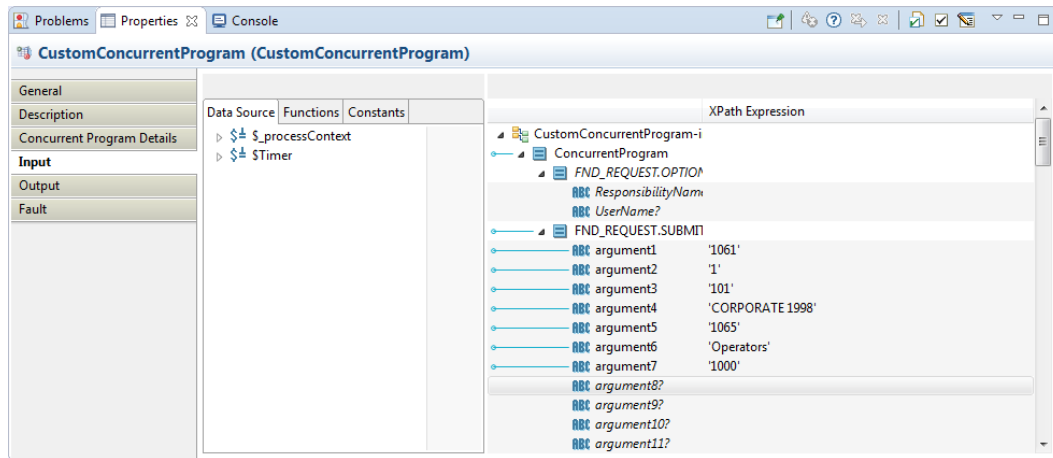
1. Create a project in TIBCO Business Studio.
2. Create a process in the project.

### Procedure

1. Add the Custom Concurrent Program activity to the process.
2. Configure the Custom Concurrent Program activity, in particular, the input of the activity.

GLBBSU requires 7 arguments. See [Checking the Type and Parameters of a Concurrent Program](#) for details about the type and parameters.

The following figure shows a sample configuration of the input:



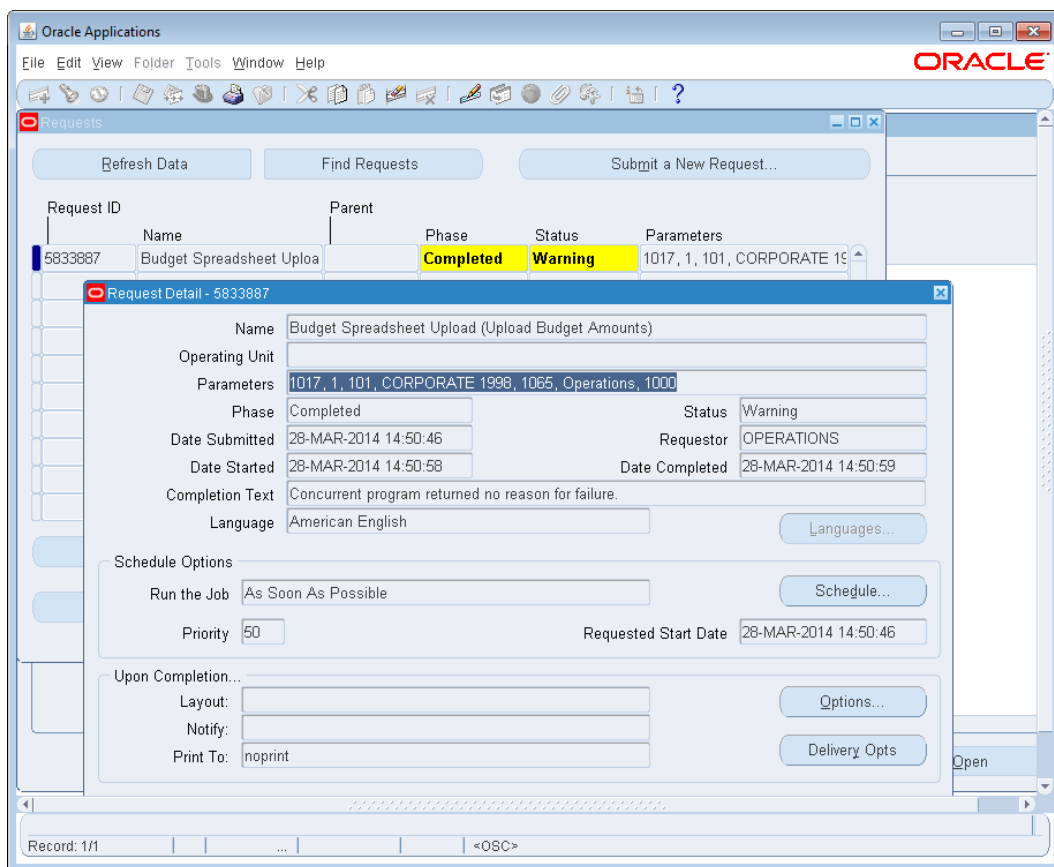
3. Run the process in Debug mode.

## Result

- Output in TIBCO Business Studio: the **STATUS** field is **Warning**, indicating that the process has run without an error, and the request is executed successfully with warnings.

CustomConcurrentProgram		
Input	Name	Value
Output	CustomConcurrentProgram	<ns0:ConcurrentProgram xmlns:ns0="http://www.tibco.com/namespaces/tnt/plugins/customconcur
	REQUEST_ID	5832314
Fault	PHASE	Completed
	STATUS	Warning
All	MESSAGE	Concurrent program returned no reason for failure.

- Request results in Oracle E-Business Suite: the following figure shows the request details in Oracle E-Business Suite.



# Introduction to Oracle Integration Repository

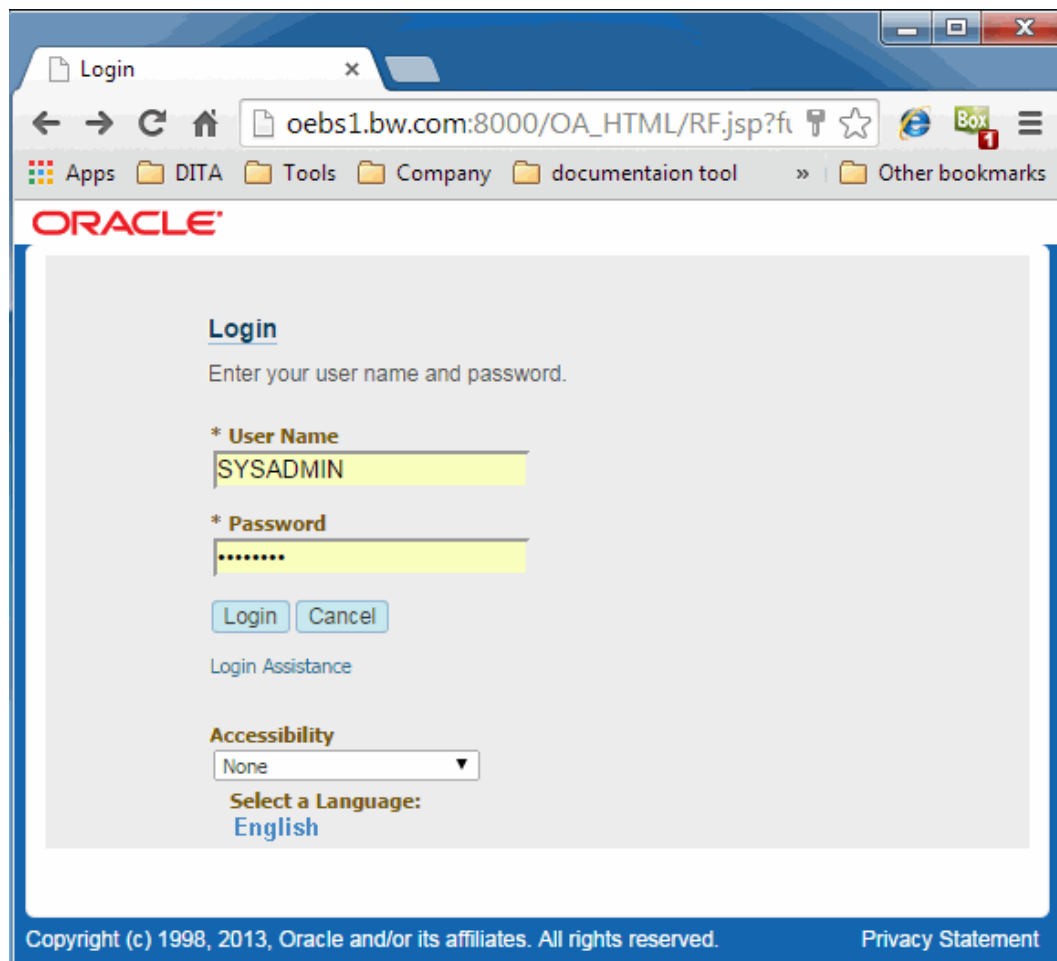
Oracle Integration Repository is a compilation of information about the service endpoints exposed by Oracle E-Business Suite of applications. It provides a complete catalog of the business service interfaces of Oracle E-Business Suite.

## Searching for a Business Service Interface in Oracle Integration Repository

You can view the details of a business service interface in Oracle Integration Repository.

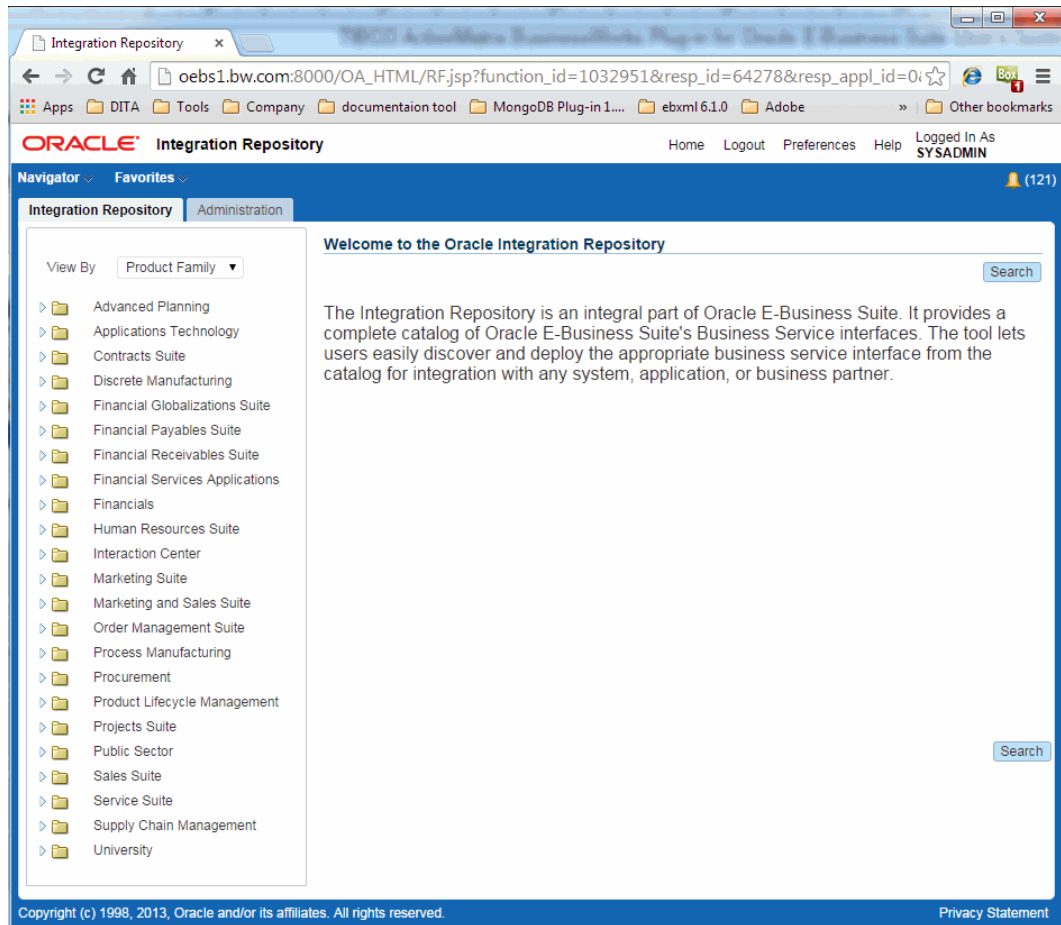
### Procedure

1. Log on to Oracle E-Business Suite using a system administrator's account.

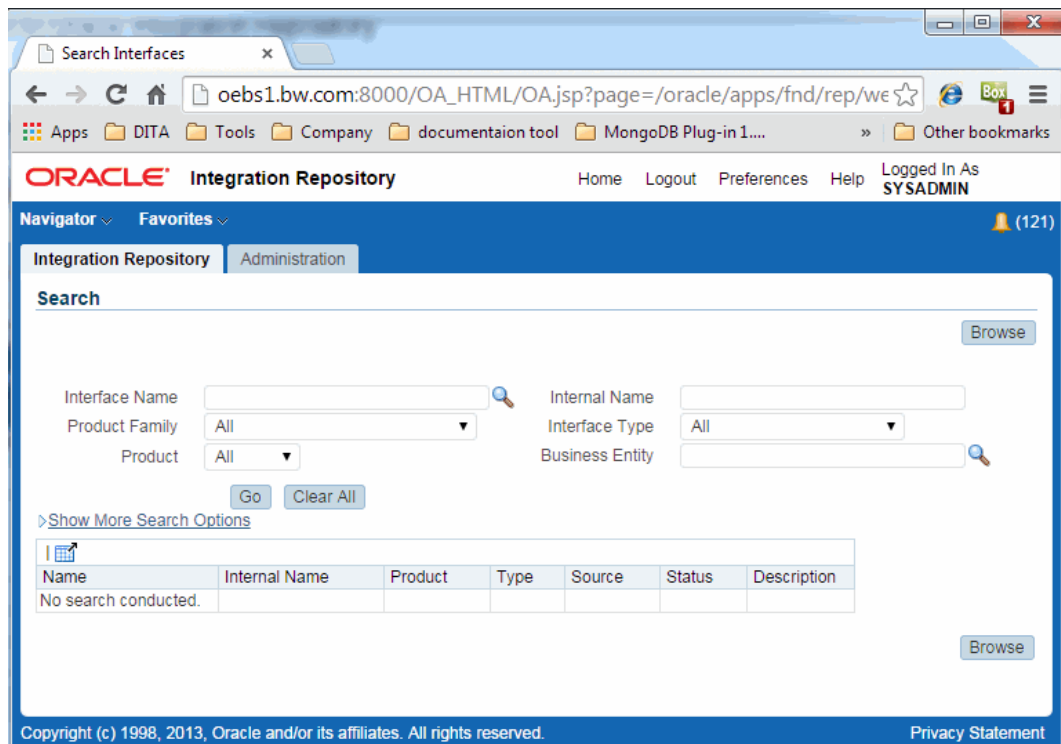


2. Expand **Integrated SOA Gateway > Integration Repository** in the Main Menu panel.  
The welcome page of Oracle Integration Repository is displayed, as shown in the following figure.





3. Click **Search** in the upper-right corner to show the search page.



4. Enter the **Internal Name** of a business service interface, select the **Interface Type** from the list, and then click **Go** to show its details.

For example, enter the Oracle API Package name BOM\_BO\_PUB in the **Internal Name** field. In the **Interface Type** field, select PL/SQL from the list. Click **Go**. The API Package BOM\_BO\_PUB is found in Oracle E-Business Suite.

Search Interfaces

oebs1.bw.com:8000/OA\_HTML/OA.jsp?page=/oracle/apps/fnd/rep/webui/InterfacesSearchPG&Re

ORACLE Integration Repository

Home Logout Preferences Help Logged In As SYSADMIN

Navigator Favorites

Integration Repository Administration

Search

Interface Name:  Internal Name: BOM\_BO\_PUB

Product Family: All Interface Type: PL/SQL

Product: All Business Entity:

Go Clear All

Show More Search Options

Export |

Name	Internal Name	Product	Type	Source	Status	Description
<a href="#">Create Add or Update Bill of Material Business Entities</a>	BOM_BO_PUB	Bills of Material	PL/SQL	Oracle	Active	APIs in this package are used to create, update or delete single or multiple components of a Structure/BOM. First, the user creates the structure header for an Item. After creating the Header the user adds or updates or deletes components (Items) and their child entities such as reference designators, substitute components etc.

Copyright (c) 1998, 2013, Oracle and/or its affiliates. All rights reserved. Privacy Statement

5. Click the name of the business service interface to show its details.

PLSQL Interface : Create, Add or Update Bill of Material Business Entities

Internal Name: BOM\_BO\_PUB

Type: PL/SQL

Product: Bills of Material

Status: Active

Scope: Public

Interface Source: Oracle

Log Configuration: Disabled

Configure

Overview SOAP Web Service REST Web Service Grants

Full Description

APIs in this package are used to create, update or delete single or multiple components of a Structure/BOM. First, the user creates the structure header for an Item. After creating the Header the user adds or updates or deletes components (Items) and their child entities such as reference designators, substitute components etc. Implementation of each of these entities are described below. This API can be used for processing of a single or multiple business entities (Structure/BOM and its child entities) per call. The entities that needs to be processed should belong to the same Structure/BOM. How to use this is explained below through examples.

Example 1: To Create a new Bom entity (Bom Header, Component, Reference Designator etc).

- 1. The User should initialize the Error\_Handler so that errors can be logged and retrieved.
- 2. The user populates the Record Type for each entity like Bill Header that needs to be processed.
- 3. The record should be created with attribute values as explained in the record type description below
- 4. For example populate the Bom Header by giving values as follows Bom\_Header.assembly\_type = 1
- 5. Then Process\_Bom procedure in this package is called with already created record types as parameters
- 6. The Process\_Bom method processes the records and registers errors in the plsql error table which can be extracted using Error\_Handler.get\_message
- 7. If the Return Status is 0 then the process completed successfully else The error is logged in the Error\_Handler
- 8. If Successful the user should commit the data.

Example-2 To Update the value of an attribute

- 1. If the user wants to update the user unique index attributes like Operation Sequence Number he should give the existing value to pick up the correct record and the new value to change.
- 2. bom\_component\_old\_operation\_sequence\_number = 1
- 3. bom\_component\_new\_operation\_sequence\_number = 2
- 4. If the user wants to update non user unique indexes then he needs to give only the new value
- 5. bom\_component\_quantity = 30

# Introduction to Oracle Database JPublisher

Oracle Database JPublisher is maintained by Oracle, and the packaging and usage are subject to change. If any of the following information is no longer valid, check with Oracle Support or contact TIBCO Support for more information.

When supplied with the correct parameters, JPublisher generates two types of outputs:

- It generates a PL/SQL script that exposes the custom data types, stored procedures and functions contained in a package by creating SQL types, and a wrapper package which makes the internals of the original package more accessible and callable.
- It generates Java classes which utilize the generated wrapper PL/SQL scripts. These wrapper Java classes model the database objects and can be used programmatically to execute the contents of the selected package.

## Wrapper and Non-Wrapper Data Types

The difference between wrapper and non-wrapper data types lies in their requirements for the use of Oracle Database JPublisher.

### Wrapper Data Types

The wrapper data types are listed as follows:

- PLSQL BOOLEAN
- TABLE
- PLSQL TABLE
- RECORD
- VARRAY

The following code block shows an example that uses some wrapper data types:

```
CREATE OR REPLACE PACKAGE WRAPPER_DATA_TYPE_PKG
AS
TYPE PER_INFO_RECORD IS RECORD
(FIRST_NAME VARCHAR2 (20)
, LAST_NAME VARCHAR2 (20)
, SEX VARCHAR2 (1))
, AGE NUMBER (3)
, PER_ID NUMBER
)
TYPE PER_TAB IS table of PER_INFO_RECORD;
PROCEDURE CREATE_PERSONS (P_PERS PER_TAB, P_COMMIT BOOLEAN);
PROCEDURE CREATE_PERSON (P_PER PER_INFO_RECORD, P_STATUS VARCHAR2);
END;
```

In this example, P\_COMMIT is PLSQL BOOLEAN type, P\_PERS is TABLE type, and P\_PER is PLSQL RECORD type.

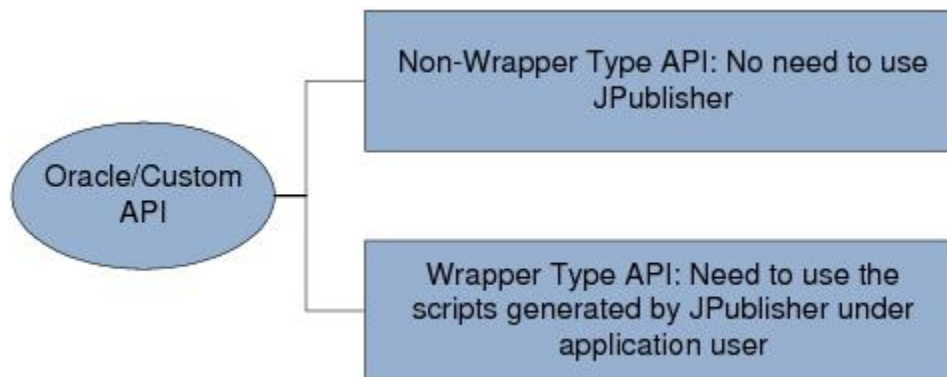
### Non-Wrapper Data Types

The non-wrapper data types are listed as follows:

- VARCHAR2
- LONG
- CHAR
- CLOB

- NUMBER
- INTEGER
- DATE
- FLOAT
- BINARY\_INTEGER
- PLS\_INTEGER

The following diagram shows the difference between wrapper and non-wrapper data types:



## Oracle JPublisher Usage in the Plug-in

TIBCO ActiveMatrix BusinessWorks Plug-in for Oracle E-Business Suite relies on Oracle JPublisher to generate PL/SQL scripts in some instances where programmatically accessing the database objects is not possible.

TIBCO ActiveMatrix BusinessWorks Plug-in for Oracle E-Business Suite does not require the generated Java classes, because the plug-in utilizes a general framework to invoke the procedures of the stored packages.

In particular, Oracle E-Business Suite APIs that require data structures defined within the API packages require the wrappers generated by Oracle JPublisher, because these data structure definitions are not accessible outside of the package.

With the wrappers, TIBCO ActiveMatrix BusinessWorks Plug-in for Oracle E-Business Suite can thus invoke the procedures contained inside the Oracle E-Business Suite APIs packages.



Only Oracle E-Business Suite APIs that contain complex types such as table types and record types require Oracle JPublisher PL/SQL wrappers.

## Installing JPublisher

Oracle JPublisher is maintained by Oracle, and the packaging and usage are subject to change. If any of the information conveyed here is no longer valid, check with Oracle Support or contact TIBCO Support for more information.

### Prerequisites

Ensure that your system meets the following requirements:

- Java JDK 1.5 or higher is installed on your system.
- The bin directory is in the path. On Windows systems, it is good practice to verify that Java is accessible by entering `java -version_number` in a command line window to reduce the chance of errors.

## Procedure

1. Download a supported version of Oracle JPublisher from the Oracle website, <http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html>.

Check whether the version of your JPublisher is supported. See [Supported Versions](#) for details.



If you want to use Oracle JPublisher 11.x, copy the `runtime12.jar` and `translator.jar` from the `ORACLE_HOME\oc4j\sql\jlib` directory to the `TIBCO_HOME\bw\palettes\oracleebs\6.1\examples\jpublisher\lib` directory.

2. Extract the package to a local folder in your system.
3. Reference `runtime12.jar`, `translator.jar`, and the JDBC library in the Java classpath.
  - On Microsoft Windows platforms:
    1. Select **Start > Control Panel > System**.
    2. In the System Properties dialog, click the **Advanced** tab, and then click **Environment Variables** in this tab.
    3. In the System variables area, select `CLASSPATH`, and click **Edit**.
    4. Add `runtime12.jar`, `translator.jar`, and the JDBC library to the **Variable value** field. Click **OK**.

- On Windows platforms with console window:

Execute the following command:

```
set
CLASSPATH=JPUB_LIB_DIR/runtime12.jar;JPUB_LIB_DIR/translator.jar;
JPUB_LIB_DIR/ojdbc5.jar
```

- On UNIX platforms:

Execute the following command:

```
export
CLASSPATH=JPUB_LIB_DIR/runtime12.jar:JPUB_LIB_DIR/translator.jar:
JPUB_LIB_DIR/ojdbc5.jar
```



It is good practice to put the JDBC library in the same folder as the `runtime12.jar` and `translator.jar` directory.

## Supported Versions

Several versions of Oracle JPublisher from Oracle website can generate compatible PL/SQL wrappers for the plug-in.

The supported versions are as follows:

- Oracle JPublisher 10.2
- Oracle JPublisher 10.1.0.3
- Oracle JPublisher 9.2.0.1



If you want to use Oracle JPublisher 11.x, copy the `runtime12.jar` and `translator.jar` from the `ORACLE_HOME\oc4j\sql\jlib` directory to the `TIBCO_HOME\bw\palettes\oracleebs\6.1\examples\jpublisher\lib` directory.

Oracle JPublisher requires JDBC libraries, which can be downloaded from the Oracle website, <http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html>.

The JDBC libraries supported by Oracle JPublisher are as follows:

- ojdbc14.jar
- ojdbc5.jar
- ojdbc6.jar



JDBC library ojdbc14.jar is for JPublisher 9.2.0.1, 10.1.x, and 10.2.x; ojdbc5.jar and ojdbc6.jar are for JPublisher 11.x.

## Using JPublisher

Oracle Database JPublisher is a command line utility that accepts a few parameters for specifying the connection, user, and script file details. Before running the utility, you must grant privileges to the required package.

### Granting Privileges to the Running Account

To run Oracle JPublisher, the logged-in user must have access to the package and all database objects related to the package for which the wrapper is generated.

The apps user typically has access to all packages. It is good practice to use the apps user to run Oracle JPublisher for script generation. If the apps user cannot be used directly, you must grant privileges to the required package to the plug-in user under the apps user.

#### Prerequisites

Identify all procedures, related packages, tables, and other items used by the package for which the wrapper is generated.

#### Procedure

Execute the following SQL statements:

```
GRANT EXECUTE ON BOM_BO_PUB TO username;  
CREATE SYNONYM BOM_BO_PUB FOR APPS.BOM_BO_PUB;
```



To run this script, the plug-in user must also have access to tables, other packages and the tables that these packages might reference, and other objects accessed by any dependent packages. Therefore, it is good practice to run this script by using the apps user; in this case, you do not have to grant privileges to the plug-in user.

## Running JPublisher

You can run JPublisher on a command line.

Open a command line window, and execute the following example command:

```
java oracle.jpub.Main -user=apps/apps -  
url=jdbc:oracle:thin:@10.105.176.40:1521:VIS -sql=BOM_BO_PUB -  
plspackage=TIB_BOM_BO_PUB -plsmap=always -dir=java/BOM_BO_PUB -plsfile=sql/  
BOM_BO_PUB/BOM_BO_PUB.sql
```

The execution might take some time (a few minutes for each PL/SQL wrapper) depending on the size of the API package specified and the database server resources available. During the execution, JPublisher lists the database types that have been wrapped.

JPublisher might attempt to compile the Java classes generated by running the Java compiler. This attempt might fail because `-plsmap=always` is selected. However, it does not affect the generation of the PL/SQL wrappers.


## Result

When JPublisher executes the command successfully, two SQL scripts are generated, including a wrapper file and a file that is used to undo the wrapper changes. See [JPublisher Wrapper Details](#) for more details.

## JPublisher Parameters

You can specify the user name used to log in to the database, the package name that is used to generate the PL/SQL wrapper, the name of the generated wrapper package, the filename of the generated PL/SQL wrapper, and the directory to save the PL/SQL wrapper file.

The following table lists the JPublisher parameters:

Parameter	Description
<b>-user</b>	<p>Specifies the credentials used to log in to the database to extract the API information.</p> <p> The logged-in user must have all privileges to the API packages and Oracle Database Dictionary tables. Therefore, it is good practice to use the apps user to avoid any privilege issues.</p>
<b>-url</b>	<p>Contains the JDBC connection string used to connect to the database.</p> <p>Typically, the format is <code>jdbc:oracle:thin:@[Database Server]:[Port]:[SID]</code>. For example, <code>jdbc:oracle:thin:@oracle-ebs-server:1521:VIS</code>.</p>
<b>-plssqlmap</b>	<p>Initializes Oracle JPublisher to generate the PL/SQL script wrapper.</p> <p>Its value is always. For example, <code>-plssqlmap=always</code>.</p>
<b>-sql</b>	<p>Specifies the package name in the Oracle database that is used to generate the PL/SQL wrapper.</p>
<b>-plssqlpackage</b>	<p>Specifies the name of the generated wrapper package.</p> <p>It is good practice to use <code>TIB_</code> as the prefix for the API package name. For example, if <code>-sql=BOM_BO_PUB</code>, then <code>-plssqlpackage=TIB_BOM_BO_PUB</code>.</p>
<b>-plssqlfile</b>	<p>Specifies the filename of the generated PL/SQL wrapper.</p> <p>It is good practice to use the SQL package name or the PLS package name (for example, <code>BOM_BO_PUB</code> or <code>TIB_BOM_BO_PUB</code>) so that the filename can be used to identify the package referenced by the PL/SQL wrapper.</p>
<b>-dir</b>	<p>Specifies the directory to save the PL/SQL wrapper file.</p>

## JPublisher Wrapper Details

After JPublisher executes the command successfully, two PL/SQL files are generated. One is the wrapper file which exposes data structure types and stored procedures for the execution through JDBC. The other file with the `drop` suffix contains scripts used to undo the changes of the wrapper, and makes the database return to its prior state.

For example, the files generated by JPublisher by executing the command in [Running JPublisher](#) are as follows:

- BOM\_BO\_PUB.sql
- BOM\_BO\_PUB\_drop.sql

Inspection of the generated wrapper file shows that the package types are mapped to general SQL types, and the package procedures and functions are mapped to identical procedures in the generated wrapper package. The Oracle E-Business Suite apps user or users with equivalent privileges can access the SQL type and wrapper package.

You can edit the wrapper file to restrict the access to certain procedures and types so that the entire package content might not be exposed. However, this operation requires extensive knowledge of PL/SQL, the API package, and API and type relationships for the particular API that is being configured. It is good practice to run the unmodified file in its entirety.

When using an SQL execution tool such as Oracle SQL Developer, you can use the plug-in user if privileges to the package have been granted to the plug-in user or simply use the apps user, and execute this script.



The execution of the wrapper files might require a SQL tool that can accommodate very long individual lines. It is good practice to use Oracle SQL Developer to execute this script, because SQL\*Plus might encounter problems when it executes lines that are very long, even after you increase the line size.



## Plug-in Error Codes

The following table lists the error codes, detailed explanation of each error, where applicable, and ways to solve different errors.

Error Code and Error Message	Role	Category	Description	Solution
TIBCO-BW-PALETTE-ORACLEEBS-510001 JMS Exception: %1	errorRole	BW-Plug-in	A JMS exception occurred. The diagnosis depends on the error message returned.	Check the JMS server status.
TIBCO-BW-PALETTE-ORACLEEBS-510002 Failed to create connection pool: %1	errorRole	BW-Plug-in	The creation of the connection pool failed.	Check the Oracle connection parameters.
TIBCO-BW-PALETTE-ORACLEEBS-510010 SQL Exception: %1	errorRole	BW-Plug-in	An SQL execution failed.	Ensure that you connect to the right database.
TIBCO-BW-PALETTE-ORACLEEBS-510011 Timeout Exception: %1	errorRole	BW-Plug-in	An SQL execution timed out.	Ensure that you connect to the right database.
TIBCO-BW-PALETTE-ORACLEEBS-510012 Load Resource Exception: %1	errorRole	BW-Plug-in	Cannot find the Shared Resource name. The Oracle Connection cannot be located.	Ensure that you select a connection resource for the activity.
TIBCO-BW-PALETTE-ORACLEEBS-510013 Exception: %1	errorRole	BW-Plug-in	General exception.	Inspect the error message to diagnose the issue.
TIBCO-BW-PALETTE-ORACLEEBS-510016 Advanced Queue connection failed: %1	errorRole	BW-Plug-in	The connection to Oracle Advanced Queue failed.	Check the connection parameters and the Oracle database.
TIBCO-BW-PALETTE-ORACLEEBS-510018 PLSQL API or Concurrent Program Execute Error: %1	errorRole	BW-Plug-in	An error occurred when executing SQL commands submitted by PLSQL API or Concurrent Program.	Inspect the error message to diagnose the error.

Error Code and Error Message	Role	Category	Description	Solution
<p>TIBCO-BW-PALETTE-ORACLEEBS-510019</p> <p>Connection to AQ server failed. Can't get connection.</p>	errorRole	BW-Plug-in	The connection to Oracle Advanced Queue failed.	Check the connection parameters and the Oracle Database.