

TIBCO ActiveMatrix BusinessWorks™ Plug-in for SmartMapper User's Guide

*Software Release 7.1
May 2016*

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 SmartMapper, TIBCO ActiveSpaces Enterprise Edition, 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 © 1999-2016 TIBCO Software Inc. All rights reserved.

TIBCO Software Inc. Confidential Information

Contents

TIBCO Documentation and Services	7
Product Overview	8
SmartMapper ER Model Overview	8
Entity	9
Relationship	10
Storage Service	11
SmartMapper Life Cycle	12
Getting Started	14
Creating a Project	14
Creating a SmartMapper ER Model	15
Creating a SmartMapper ER Model Shared Resource	15
Creating Applications and Entities	16
Defining Relationships and Participants	16
Adding Storage Services	16
Configuring a Process	17
Testing a Process	18
Deploying an Application	19
TIBCO Business Studio Overview	19
SmartMapper ER Model Shared Resource	21
SmartMapper ER Model	21
Entity	21
Relationship	23
Participant	24
Storage Service	24
JDBC-Based SmartMapper Service	24
Synchronizing SmartMapper ER Model	28
Sync To DB Tab	29
Reload From DB Tab	29
Adding ActiveSpaces Library into Environment Variables	30
File-Based SmartMapper Service	31
SmartMapper Palette	32
Bulk Extract	32
Bulk Extract Activity - General Tab	32
Bulk Extract Activity - Description Tab	33
Bulk Extract Activity - Input Tab	33
Extraction Filter	34

Bulk Extract Activity - Output Tab	35
Bulk Extract Activity - Fault Tab	38
Bulk Load	38
Bulk Load Activity - General Tab	38
Create Mode Setting	39
Add Mode Setting	40
Update Mode Setting	41
Bulk Load Activity - Description Tab	42
Bulk Load Activity - Input Tab	42
Bulk Load Activity - Output Tab	45
Bulk Load Activity - Fault Tab	45
Create Entity Instance	45
Create Entity Instance Activity - General Tab	45
Create Entity Instance Activity - Description Tab	45
Create Entity Instance Activity - Input Tab	45
Create Entity Instance Activity - Output Tab	46
Create Entity Instance Activity - Fault Tab	46
Update Entity Instance	46
Update Entity Instance Activity - General Tab	46
Update Entity Instance Activity - Description Tab	46
Update Entity Instance Activity - Input Tab	47
Update Entity Instance Activity - Output Tab	47
Update Entity Instance Activity - Fault Tab	47
Delete Entity Instance	48
Delete Entity Instance Activity - General Tab	48
Delete Entity Instance Activity - Description Tab	48
Delete Entity Instance Activity - Input Tab	48
Delete Entity Instance Activity - Output Tab	48
Delete Entity Instance Activity - Fault Tab	48
Create Relationship Instance	49
Create Relationship Instance Activity - General Tab	49
Create Relationship Instance Activity - Description Tab	49
Create Relationship Instance Activity - Input Tab	49
Create Relationship Instance Activity - Output Tab	49
Create Relationship Instance Activity - Fault Tab	49
Add to Relationship Instance	50
Add to Relationship Instance Activity - General Tab	50
Add to Relationship Instance Activity - Description Tab	50
Add to Relationship Instance Activity - Input Tab	50

Add to Relationship Instance Activity - Output Tab	51
Add to Relationship Instance Activity - Fault Tab	51
Delete Relationship Instance	51
Delete Relationship Instance Activity - General Tab	51
Delete Relationship Instance Activity - Description Tab	52
Delete Relationship Instance Activity - Input Tab	52
Delete Relationship Instance Activity - Output Tab	52
Delete Relationship Instance Activity - Fault Tab	52
Lookup	53
Lookup Activity - General Tab	53
Lookup Activity - Description Tab	53
Lookup Activity - Input Tab	53
Lookup Activity - Output Tab	54
Lookup Activity - Fault Tab	54
Dynamic Lookup	54
Dynamic Lookup Activity - General Tab	55
Dynamic Lookup Activity - Description Tab	55
Dynamic Lookup Activity - Input Tab	55
Dynamic Lookup Activity - Output Tab	56
Dynamic Lookup Activity - Fault Tab	56
Dynamic Delete	56
Dynamic Delete Activity - General Tab	56
Dynamic Delete Activity - Description Tab	58
Dynamic Delete Activity - Input Tab	58
Dynamic Delete Activity - Output Tab	59
Dynamic Delete Activity - Fault Tab	59
Generate Key	60
Generate Key Activity - General Tab	60
Generate Key Activity - Description Tab	60
Generate Key Activity - Output Tab	60
Generate Key Activity - Fault Tab	60
SmartMapper Wizard	60
SmartMapper Wizard Activity - General Tab	61
SmartMapper Wizard Activity - Description Tab	61
SmartMapper Wizard Activity - SmartMapper Calls Tab	61
Invoke SmartMapper Operations	62
SmartMapper Wizard Activity - Input Editor Tab	65
SmartMapper Wizard Activity - Input Tab	65
SmartMapper Wizard Activity - Output Tab	65

SmartMapper Wizard Activity - Fault Tab	65
SmartMapper Tools	67
Configuring Database Connection	67
Using Bulk Extract Tool	68
Using Bulk Load Tool	68
Data Format	69
Using Differencing Tool	69
Using Migration Tool	70
Using Database Upgrade Tool	71
Parameters for SmartMapper Tools	71
Working with Sample Projects	75
Importing Sample Projects	75
Working with Getting_Started Project	76
Business Scenario	76
SmartMapper ER Model	76
Configuration for Getting_Started Project	77
Running Getting_Started Application	78
Working with Wizard Project	79
Business Scenario	79
SmartMapper ER Model	80
Configuration for Wizard Project	81
Running Wizard Application	82
Working with BulkExtractFilter Project	83
SmartMapper ER Model	83
Configuration for BulkExtractFilter Project	83
Running BulkExtractFilter Application	84
Managing Logs	86
Log Levels	86
Setting Up Log Levels	86
Exporting Logs to a File	87
Error Codes	88

TIBCO Documentation and Services

Documentation for this and other TIBCO products is available on the TIBCO Documentation site. This 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 at <https://docs.tibco.com/products/tibco-activematrix-businessworks-smartmapper-plugin>. To directly access documentation for this product, double-click the following file:

`TIBCO_HOME/release_notes/TIB_adsmartmapper_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\bw6`. On UNIX systems, the default `TIBCO_HOME` is `/opt/tibco`.

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

- *TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper Installation*
- *TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper User's Guide*
- *TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper BUI User's Guide*
- *TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper Release Notes*

The following documents provide additional information and can be found in the TIBCO Documentation Library:

- *TIBCO ActiveMatrix BusinessWorks Documentation*
- *TIBCO Enterprise Administrator User's Guide*

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 TIBCO Community

TIBCO Community 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. TIBCO Community offers forums, blogs, and access to a variety of resources. To register, go to the following web address:

<https://community.tibco.com>

Product Overview

With TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper, you can manage and maintain cross-referencing data across applications.

TIBCO ActiveMatrix BusinessWorks is an easy-to-use integration product suite for enterprise, web, and mobile applications. It uses the Eclipse graphical user interface (GUI), TIBCO Business Studio, for defining business processes and the process engine to execute them.

TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper plugs into TIBCO ActiveMatrix BusinessWorks. Using this plug-in, you can model, assemble, deploy, and manage relationships. The plug-in also provides a unique set of features that both simplify and enhance the process of looking up and synchronizing cross-referencing data.

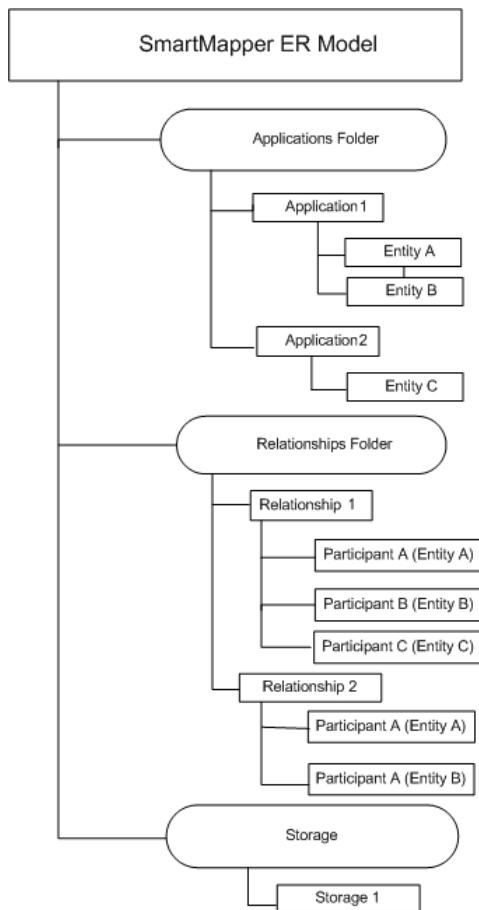
The main features of the plug-in are as follows:

- **SmartMapper ER Model**
You can model your own relationships as well as specify a storage service where cross-referencing data is stored. See [SmartMapper ER Model Overview](#) for more details.
- **SmartMapper Activities**
You can use these activities to create, update, delete, and look up relationships as well as to load and extract cross-referencing data in bulk. See [SmartMapper Palette](#) for more details.
- **SmartMapper Tools**
You can use the tools to load and extract cross-referencing data in bulk, upgrade databases, migrate cross-referencing data to an empty database, and compare two ER Models. See [SmartMapper Tools](#) for more details.
- **SmartMapper Business UI**
You can use it to maintain cross-referencing data in a web-based application. See *TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper BUI User's Guide* for more details.

SmartMapper ER Model Overview

A SmartMapper ER Model defines a set of entities and the relationships where the entities participant in.

An entity-relationship model (ER model) is a systematic way of describing and defining business process. A SmartMapper ER Model represents a logical data model that contains a set of applications, relationships, and storage services.



As shown in the previous figure, a SmartMapper ER Model contains three basic components:

- Applications

An **Applications** folder contains one or multiple applications as well as subordinate folders. An *application* must contain at least one *entity*. An application groups the entities based on logical systems.

- Relationships

A **Relationships** folder contains one or more relationships as well as subordinate folders. A *relationship* is a collection of entities that you want to bind together in a particular way, such as a sales relationship between sales orders and accounts.

- Storage Services

A **Storages** folder contains one or more storage services. A *storage service* specifies whether the data is stored in a file or a database.

Entity

An entity represents something that is being referenced.

An entity is a proxy for an object. It contains necessary information to retrieve or identify the object. Entities are created and grouped logically in application folders like SAP, Oracle, and Siebel. Once created, entities are placed in relationships to form *participants*.

An entity represents a complete key to the object, which is composed of one or more attributes. Non-key attributes are supported. Certain non-key attributes are also nullable. If an entity has multiple attributes marked as keys, they form a composite key.

Attributes	Description
Key	Attribute(s) to define an entity. A key is used to retrieve and uniquely identify a set of information. A key is either a single attribute or a set of attributes (i.e. composite key), such attributes cannot be null or blank.
Primary Key	The unique key to an object. The most concise way to uniquely identify an object. The primary key to an object must be one (or more) of the attributes of the entity which represents the object.
Composite Key	A composite key is used when an entity needs more than one attribute for unique identification.
Foreign Key	A foreign key represents an object other than the object that contains the key.

Relationship

A relationship captures how entities are related to each other. Two relationship types are available in TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper, association and identity.

A participant is the role that an entity plays in a relationship. In TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper, the role that an entity plays is that of itself.

The following two relationship types are used in the plug-in:

- [Association Relationship](#)
- [Identity Relationship](#)

Association Relationship

An association relationship has two and only two participants. They are used to model a relationship of two typically orthogonal entities, like teachers and classes, as shown in the table, or a relationship of the same entity, like a part-subpart containment relationship.

An association relationship is quite different from an identity relationship. An association relationship describes a relationship between participants that belong to two different kinds like customer and order; while an identity relationship describes a relationship of participants that belong to a same kind from different systems like the order ID in different systems.

Teacher	Class (Grade-Class Number)
Jody	2-1
Jody	2-2
Dover	2-1
Dover	1-1
McDonald	1-2

Identity Relationship

All participants are peers in an identity relationship. It is used to keep the corresponding keys of different applications. For example, after a company merger involving several applications, you are given the task to create a cross-reference of keys for three data file dumps, one for SAP, one for Siebel, and one for Oracle.

The following table indicates the simplest case, that is, a customer record is displayed in each application once and only once (the relationship cardinality is 1-1-1).

SAP_Customer	Siebel_Account	Oracle_Customer
A	1	1234 J
B	2	1222 K
C	3	1433 L
D	4	4123 M

When a customer record is not in all the applications, and the record is in one application, it can be found only once (the relationship cardinality is 1-1-1).

SAP_Customer	Siebel_Account	Oracle_Customer
NULL	1	1234 J
B	2	1222 K
C	3	1433 L
D	4	4123 M

When one customer record is not displayed or is displayed more times in the applications, the identity relationship is as follows (the relationship cardinality is M-M-M).

SAP_Customer	Siebel_Account	Oracle_Customer
NULL	1	1234 J
B	NULL	1222 K
		1122 M
C	3	NULL
F	5	
D	4	4123 M
E		

Storage Service

A storage service defines the back-end implementation and storage characteristics of a SmartMapper ER Model.

A SmartMapper ER Model is only assigned to one storage service at a time. If you assign a SmartMapper ER Model, which is already assigned to one service, to another service, the SmartMapper ER Model is automatically un-assigned from the first service and assigned to the new service.

The following storage services are available:

- [JDBC-Based SmartMapper Service](#)
- [File-Based SmartMapper Service](#)

JDBC-Based SmartMapper Service

With the JDBC-based SmartMapper service, you can store cross-referencing data in a database. Microsoft SQL Server, Oracle and MySQL are supported.

Currently, the driver of Microsoft SQL Server is packaged with BusinessWorks. If you want to use Oracle and MySQL, you are required to add the drivers for them both at design time and run time, see *TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper Installation* for more information.

File-Based SmartMapper Service

The file-based SmartMapper service is very useful for prototyping and for production where the data is static. But you cannot write to the files, such as for seeding purposes, it is very slow. In addition, there is some risk of data corruption when doing concurrent writes.

While the JDBC-based service accesses a database, the file-based services do not. Instead, the file-based services are 100% cached. The file seeds the cache. Writes might be done for seeding and maintaining the file, but this is not a substitute for a database where there are large volumes of writes.

The file-based service runs inside the TIBCO ActiveMatrix BusinessWorks process engine and reads and writes the SmartMapper data in an XML file.

The file-based service does not perform well in write-intensive applications. The integrity of a file-based data store is lower than a database. File-based storage cannot be used for data that cannot be regenerated.

SmartMapper Life Cycle

TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper provides an easy-to-use graphical user interface to maintain cross-referencing relationships for various applications.

A life cycle of TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper is composed with the following four phases:

1. Model Phase

In this phase, you build your own relationships between applications by [creating a SmartMapper ER Model](#).

Each SmartMapper ER Model is assigned to one persistent storage. If multiple SmartMapper ER Models are created, they must be separated. That is, a relationship only refers to entities in its own SmartMapper ER Model.

2. Assemble Phase

In this phase, you assemble your relationships that are created in the model phase. That is, [configure a business process](#).

There are two ways to configure a process, using individual activities in [SmartMapper Palette](#), or using the [SmartMapper Wizard](#), which is useful in case of a large number of lookups.

3. Deployment Phase

In this phase, you attach the BusinessWorks process to a physical architecture.

You can [deploy your BusinessWorks application](#) by using an ActiveMatrix BusinessWorks engine which is configured with SmartMapper activities.

4. Management Phase

In this phase, you manage your relationship data by using SmartMapper Business UI, which is a web-based application. See *TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper Business UI User's Guide* for more details.

Getting Started

This tutorial is designed for the beginners who want to use TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper in TIBCO Business Studio.

All the 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 SmartMapper includes:

1. [Creating a Project](#)
2. [Creating a SmartMapper ER Model](#)
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.

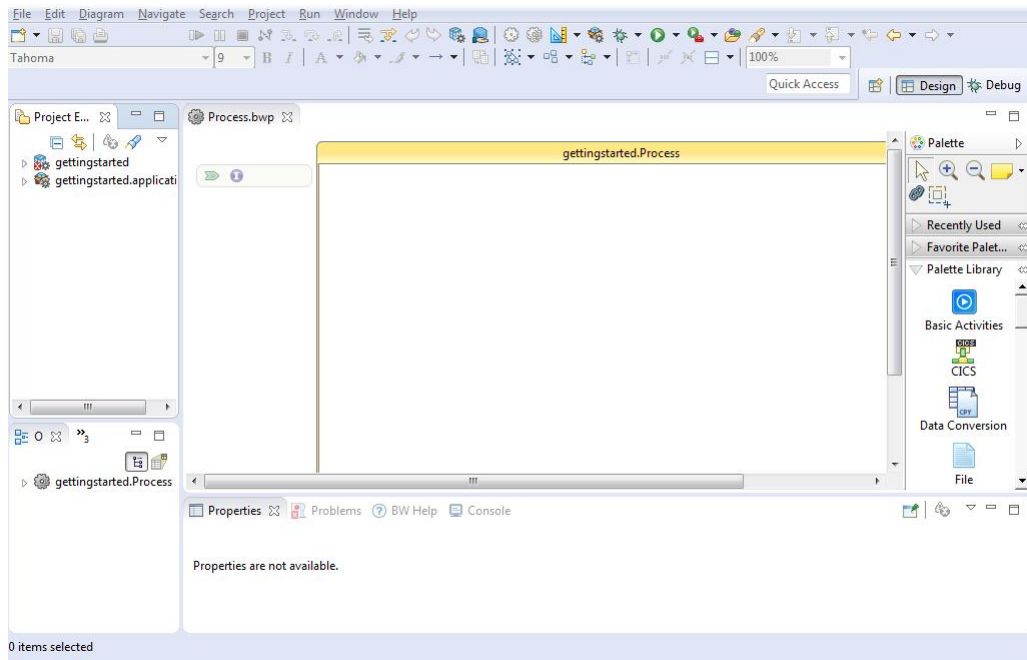
An Eclipse project is an application module configured for TIBCO ActiveMatrix BusinessWorks. An application module is the smallest unit of resources that is named, versioned, and packaged as part of an application.

Procedure

1. Start TIBCO Business Studio 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 the Java tooling capabilities.
 - f) Click **Finish** to create the project.

Result

The project with the specified settings is displayed in the Project Explorer view.



Creating a SmartMapper ER Model

A SmartMapper ER Model is needed before configuring a process. A SmartMapper ER Model is a reusable logical data model that contains a set of applications, entities, relationships, and storage services.

To create a complete SmartMapper ER Model, finish the following tasks:

1. [Creating a SmartMapper ER Model](#)
2. [Creating Applications and Entities](#)
3. [Defining Relationships and Participants](#)
4. [Adding Storage Services](#)

Creating a SmartMapper ER Model Shared Resource

A SmartMapper ER Model shared resource assembles applications, relationships, and storage services together.

Prerequisites

The SmartMapper ER Model is available at the Resources level. Before creating a SmartMapper ER Model, you must [create a project](#).

Procedure

1. Expand the created project in the Project Explorer view.
2. Right-click the **Resources** folder and click **New > SmartMapper ER Model**.
3. Type a name for the SmartMapper ER Model in the ERModel dialog. Click **Finish**.
4. Configure the SmartMapper ER Model in the displayed editor. See [SmartMapper ER Model](#) for more details on configurations.

Creating Applications and Entities

An application is a container of entities. An entity is an object that is being referenced.

Prerequisites

Ensure that you have [created a SmartMapper ER Model shared resource](#) before creating applications and entities.

Procedure

1. Click the **Applications** tab in the SmartMapper ER Model editor.
2. Optional: Click **Add Application Folder** and type a folder name in the displayed **Configuration** tab.
3. Click **Add Application** and type an application name in the **Configuration** tab.
4. Click **Add Entity** and configure the entity in the displayed **Configuration** tab.
See [Entity](#) for more details.


Defining Relationships and Participants

A relationship is a collection of entities that you want to bind together.

Prerequisites

Ensure that you have [created applications and entities](#) before defining relationships and participants.

Procedure

1. Click the **Relationships** tab in the SmartMapper ER Model editor.
2. Optional: Click **Add Relationship Folder** and type a folder name in the displayed **Configuration** tab.
3. Click **Add Relationship** and type a relationship name in the displayed **Configuration** tab.
4. Select the relationship type from **Relationship Type** list.
See [Relationship](#) for more details about relationship types.
5. Click **Add Participant** to add a participant to the relationship.
6. In the **Configuration** tab, click  to select an entity for the added participant.
7. If the relationship is an identity relationship, the **Cardinality is many** check box is displayed. You can select it or not based on your requirements.

Adding Storage Services

A storage service defines the back-end implementation and storage characteristics of a SmartMapper ER Model. Take the JDBC-Based SmartMapper Service as an example.

Prerequisites

Ensure that you have [created a SmartMapper ER Model shared resource](#) before adding storage services.

Procedure

1. Click the **Storages** tab in the SmartMapper ER Model editor.
2. Click **Add JDBC Service** and configure the storage service in the displayed **Configuration** tab.

3. Select the **Active** check box to set this storage service as active.



Only one storage service can be active among many.

4. In the **JDBC Settings** tab, select a Database type from the **Database** list, and provide correct parameters for each field.

See [JDBC Settings Tab](#) for more details.

5. Click **Test Connection** to validate the connection.
6. Click **Sync ER Model w/DB** to perform synchronization.

See [Synchronizing SmartMapper ER Model](#) for more details.

Configuring a Process

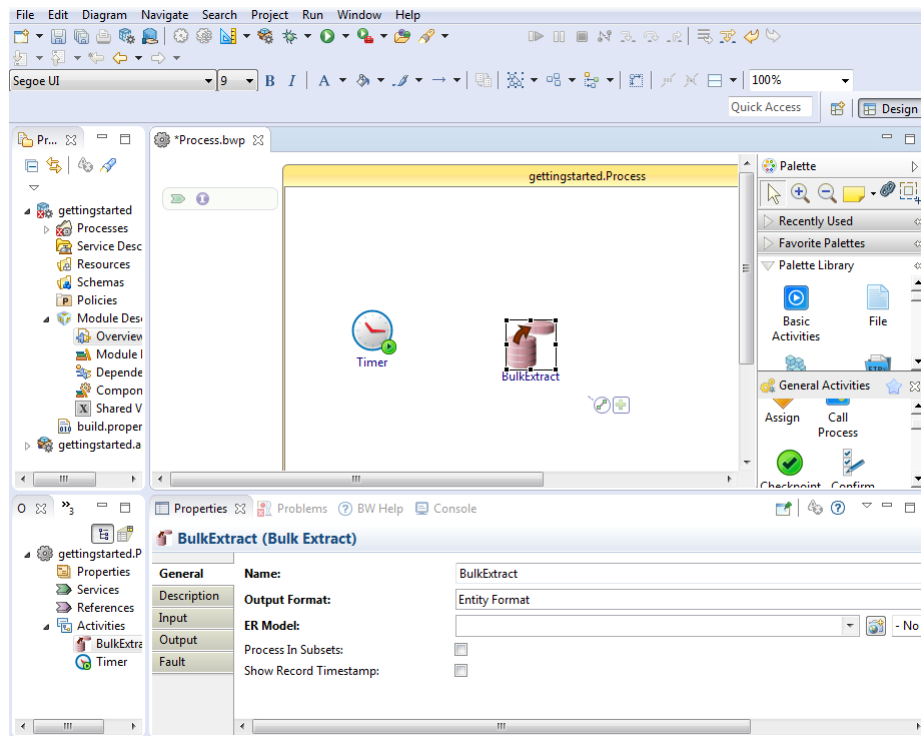
After creating a project, an empty process is created. You can add activities to the empty process to complete a task. For example, bulk extract cross-referencing data.


Prerequisites

Ensure that you have created an empty process when [Creating a Project](#).

Procedure

1. In the Project Explorer view, click the created project and open the empty process from the **Processes** folder.
2. Select an activity from the Palette view and drop it in the Process editor.
For example, select and drop the Timer activity from the General Activities palette and the Bulk Extract activity from the SmartMapper palette.



3. Drag the  icon to create a transition between the added activities.
4. Configure the added Bulk Extract activity, as described in [SmartMapper Palette](#).



A SmartMapper ER Model is required when configuring the SmartMapper activities. See [Creating a SmartMapper ER Model](#) for more details about how to create a SmartMapper ER Model.

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

Testing a Process

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

Prerequisites

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

Procedure

- On the toolbar, click **Debug > Debug Configurations**.
- Click **BusinessWorks Application > BWApplication** in the left panel.
By default, all the 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.
- Click **Debug** to test the process in the selected application.
TIBCO Business Studio changes to the Debug perspective. The debug information is displayed in the Console view.

```

[19:35:24.373 INFO [main] com.tibco.thor.frwk - TIBCO ActiveMatrix BusinessWorks version 6.3.1, build V12, 2016-01-27

<@BWEclipseAppNode> Apr 06, 2016 7:35:49 PM org.springframework.beans.TypeConverterDelegate findDefaultEditor
WARNING: PropertyEditor [com.sun.beans.editors.EnumEditor] found through deprecated global PropertyEditorManager fallback - consider using a more isolated
Apr 06, 2016 7:35:50 PM org.springframework.beans.TypeConverterDelegate findDefaultEditor
WARNING: PropertyEditor [com.sun.beans.editors.EnumEditor] found through deprecated global PropertyEditorManager fallback - consider using a more isolated
Apr 06, 2016 7:35:50 PM org.springframework.beans.TypeConverterDelegate findDefaultEditor
WARNING: PropertyEditor [com.sun.beans.editors.EnumEditor] found through deprecated global PropertyEditorManager fallback - consider using a more isolated
Apr 06, 2016 7:35:50 PM org.springframework.beans.TypeConverterDelegate findDefaultEditor
WARNING: PropertyEditor [com.sun.beans.editors.EnumEditor] found through deprecated global PropertyEditorManager fallback - consider using a more isolated
Apr 06, 2016 7:35:50 PM org.springframework.beans.TypeConverterDelegate findDefaultEditor
WARNING: PropertyEditor [com.sun.beans.editors.EnumEditor] found through deprecated global PropertyEditorManager fallback - consider using a more isolated
Apr 06, 2016 7:35:50 PM org.springframework.beans.TypeConverterDelegate findDefaultEditor
WARNING: PropertyEditor [com.sun.beans.editors.EnumEditor] found through deprecated global PropertyEditorManager fallback - consider using a more isolated
19:35:51.935 INFO [main] com.tibco.bw.frwk.engine.BWEngine - TIBCO-BW-FRWK-300002: BW Engine [Main] started successfully.
19:35:53.361 INFO [main] com.tibco.thor.frwk.Deployer - Started by BusinessStudio.
19:35:54.504 INFO [Framework Event Dispatcher: Equinox Container: 50c3335c-69fc-0015-1a4d-d35a141fe87a] com.tibco.thor.frwk.Deployer - TIBCO-THOR-FRWK-300001:
19:35:54.817 INFO [Framework Event Dispatcher: Equinox Container: 50c3335c-69fc-0015-1a4d-d35a141fe87a] com.tibco.thor.frwk.Application - TIBCO-THOR-FRWK-300001:
19:35:54.957 INFO [Framework Event Dispatcher: Equinox Container: 50c3335c-69fc-0015-1a4d-d35a141fe87a] com.tibco.thor.frwk.Application [test11.application:1.0] is impaired.
19:35:54.971 INFO [Job_Executor] com.tibco.thor.frwk.Application - TIBCO-THOR-FRWK-300001: All application dependencies are resolved for Application [test11.application:1.0]
19:35:55.941 INFO [Thread-11] com.tibco.thor.frwk.Application - TIBCO-THOR-FRWK-300006: Started BW Application [test11.application:1.0]
19:36:03.168 INFO [Thread-11] com.tibco.thor.frwk.Application - TIBCO-THOR-FRWK-300006: Started BW Application [test11.application:1.0]
  
```

- In the **Debug** tab, 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.

Name	Value
Input	BulkExtract
Output	entity
Fault	entity [1]
All	name Entity_A createTime 2016-04-06T18:33:47.283 updateTime 2016-04-06T18:33:47.283 field [1] field [1] <field>\r\n <index>1</index>\r\n <name>id</name>\r\n <value>001</value>\r\n indi 1 nan id valu 001 lastSubset false

Deploying an Application

After testing, 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 that is created in TIBCO Business Studio. For more information about TIBCO Business Studio, see [TIBCO Business Studio Overview](#).

Deploying an application involves the following tasks:

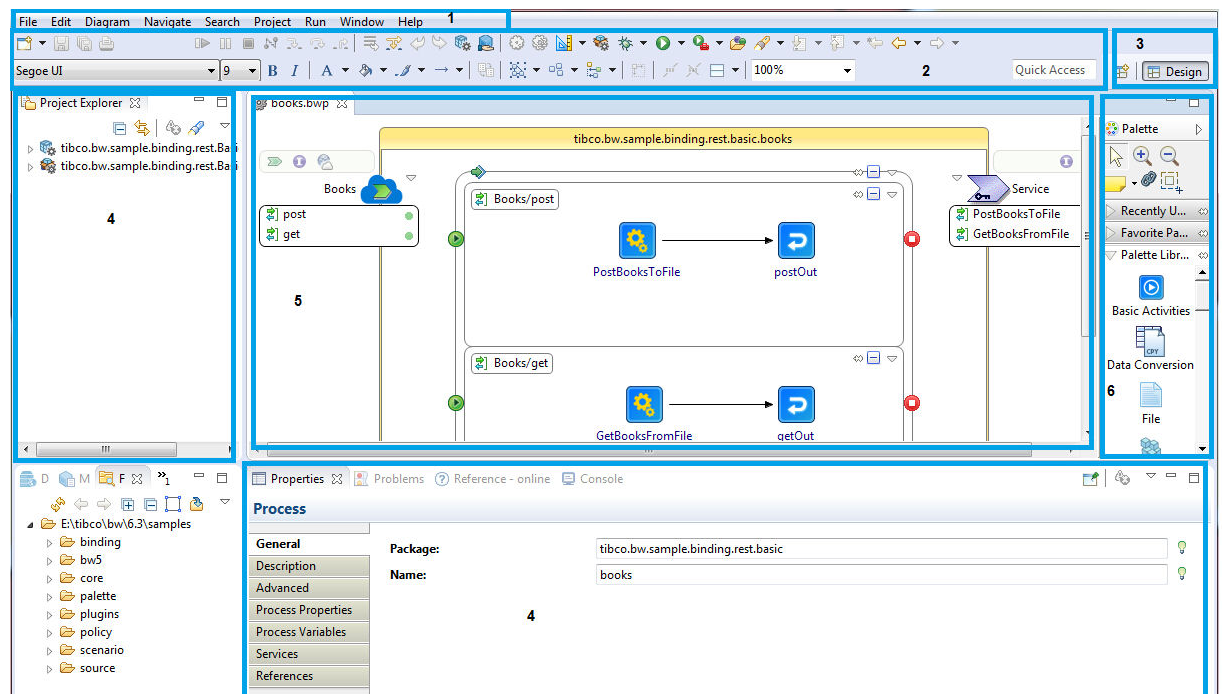
1. Uploading an application archive
2. Deploying an application archive
3. Starting an application

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:

1. **Menu:** contains menu items such as File, Edit, Diagram, Navigate, Search, Project, Run, Window, and Help.
2. **Toolbar:** contains buttons for frequently used commands such as New , Save , Enable/Disable Business Studio Capabilities , Create a new BusinessWorks Application Module , Create a new BusinessWorks Shared Module , Debug , Run , and so on.

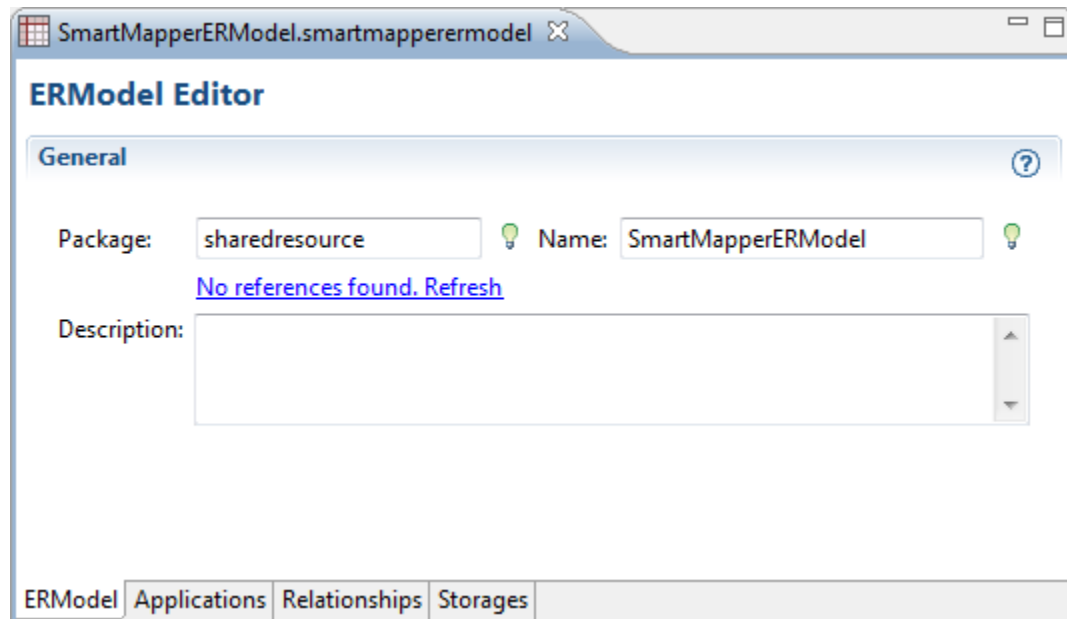
3. **Perspective:** contains an initial set and layout of views that are required to perform a certain task. TIBCO Business Studio launches the Modeling perspective by default. You can change the perspective from the menu **Window > Open Perspective > Perspective_Name**.
4. **View:** displays resources. 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 (`MortgageAppConsumer.bwp`) 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.

SmartMapper ER Model Shared Resource

With a SmartMapper ER Model shared resource, you can assemble your own data model.

As described in [SmartMapper ER Model Overview](#), the three basic components of SmartMapper ER Model are included in this shared resource.

See [Creating a SmartMapper ER Model](#) regarding how to create a SmartMapper ER Model.



SmartMapper ER Model

The SmartMapper ER Model defines a set of relationships and their participating entities.

General

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

Field	Module Property?	Description
Package	No	Name of the package where the new shared resource is added.
Name	No	The name to be displayed as the label for the resource.
Description	No	Provide a short description for this activity.

Entity



An entity represents an object that is being referenced, such as a customer. An entity contains the information necessary to retrieve or identify the object.

Entities are composed of one or more attributes. Entities are created and grouped logically in applications. Once created, entities are placed in relationships to form participants.

An entity represents a complete key to the object. This is the case whether the entity contains one or many attributes, and whether those attributes are keys or values.


Configuration Tab


The **Configuration** tab contains the following fields.

Field	Module Property?	Description
Name	No	<p>The name to be displayed as the label for the entity in the process.</p>  <p>Only digits, alphanumeric characters, and underscores can be contained in the entity name. Besides, a digit cannot be used as the first character of the entity name.</p>
Description	No	Add a short description for the entity.
TIBCO-Managed Entity	No	<p>When selected, if you do not provide a value for the TIBCO-Managed Entity, BusinessWorks process generates a key for the added entity.</p> <p>The Generate Key activity generates a key for a TIBCO-Managed entity. See Generate Key for more details.</p>  <p>When you select the TIBCO-Managed Entity check box, the key schema has only one attribute ID, which must be a string of length less than 100 characters, as in id(100).</p>

Schema Tab

The **Schema** tab contains the following fields.

Field	Module Property?	Description
Name	No	<p>The name of the attribute.</p> <p>Some database implementations have reserved words, which cannot be used as column names and attributes. For example, if you are using Microsoft SQL Server, do not name an attribute key. If an attribute is a named key, an error is returned when synchronizing the schema.</p>  <p>Only digits, alphanumeric characters, and underscores can be contained in the entity name. Besides, a digit cannot be used as the first character of the attribute name.</p>
Type	No	<p>Specify an attribute type, String or Integer.</p> <p>The string length becomes the actual length in the database. Therefore, if you are using double-byte data, longer strings are required.</p>

Field	Module Property?	Description
Length	No	Specify the maximum length of the key. This applies only to the String type.
Key	No	When selected, the attribute is part of the participant's key.
Is Required	No	<p>When selected, the attribute is required of the entity.</p> <div>  <ul style="list-style-type: none"> This check box is displayed only when the Key check box is not selected. When both Key and Is Required are not selected, the attribute is optional. </div>


Relationship

A relationship shared resource defines a relationship and identifies whether it is an identity or association relationship.

A relationship is the way to describe the binding of entities. An entity might participate in many relationships within one SmartMapper ER Model, but an entity cannot play two roles in the same relationship.

Configuration Tab

The **Configuration** tab contains the following fields.


Field	Module Property?	Description
Name	No	<p>The name to be displayed as the label for the relationship in the process.</p> <div>  <p>Only digits, alphanumeric characters, and underscores can be contained in the relationship name. Besides, a digit cannot be used as the first character of the relationship name.</p> </div>
Description	No	Add a short description for the relationship.
Relationship Type	No	<p>Select from the following relationship types:</p> <ul style="list-style-type: none"> Identity : there are one or multiple participants per identity relationship, where one of the participants acts as a primary object. A primary object exists independently of other objects. Association: there are only two participants per association relationship. <p>See Relationship for details about relationship types.</p>

Participant

A participant is a member of a relationship. The participant associates an entity to a relationship and specifies the entity's cardinality within that relationship.

Configuration Tab

The **Configuration** tab contains the following fields.

Field	Module Property?	Description
Name	No	The name to be displayed as the label for the participant in the relationship.
Cardinality is many	No	<p>When selected, this entity is on the many side of a one-to-many relationship.</p> <p> For an association relationship, the cardinality on both sides is assumed to be many.</p> <p>When cleared, this entity has a one-to-one relationship.</p> <p>Note: This option is available only when the relationship type is Identity.</p>

Storage Service

A storage service defines the back-end implementation and storage characteristics of a SmartMapper ER Model.

A storage service specifies whether the data is stored in a file or a database, see [Storage Service](#) for more details.

The following two storage services are available:

- [JDBC-Based SmartMapper Service](#)
- [File-Based SmartMapper Service](#)

JDBC-Based SmartMapper Service

The JDBC-Based Service accesses a JDBC database to manage cross-referencing data. This service runs inside a TIBCO ActiveMatrix BusinessWorks engine and uses a JDBC driver to read and write data to a database.

Three databases are supported, MySQL, SQL Server, and Oracle. Besides if you want to use MySQL or Oracle, you are required to add the official drivers. See *TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper Installation* for more details.



If you assign two (or more) SmartMapper ER Models to the same database, you are required to configure each to use a different user account to prevent the tables from collision.


Configuration Tab

The **Configuration** tab contains the following fields.

Field	Module Property?	Description
Name	No	The name to be displayed as the label for the storage service.
Active	No	Select the check box to set this storage service as active, only one storage service can be active among many.

JDBC Settings Tab



The **JDBC Settings** tab contains the following fields.





Field	Module Property?	Description
Database	No	The database type. Select the database that you want to use as the storage from the list.
JDBC Driver Class	Yes	<p>The fully-qualified java class name of the JDBC driver. The class name is given automatically when you select your database type.</p> <div>  <p>Only three driver names are supported in JDBC-Based SmartMapper Service:</p> <ul style="list-style-type: none"> For MySQL databases: <code>com.mysql.jdbc.Driver</code> For Oracle databases: <code>oracle.jdbc.OracleDriver</code> For SQLServer databases: <code>com.microsoft.sqlserver.jdbc.SQLServerDriver</code> </div>
JDBC URL String	Yes	The appropriate URL string for the driver.
User	Yes	The database user name.
Password	Yes	The database password.
Connection Pool Size	Yes	This value determines the maximum number of connections available in the pool. TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper database access always uses connection pooling.
Login Timeout (sec)	Yes	The amount of time in seconds before the login attempt times out.
Test Connection	No	The validation of the database connection. Click Test Connection to validate the specified database connection.

Field	Module Property?	Description
Sync ER Model w/DB	No	<p>The synchronization process creates new database tables. The database user specified in the User field needs to have 'create table' privileges.</p> <p>At run time, the SmartMapper activity does not create any new database tables and hence you are not required to have those privileges for running the activities.</p> <p>Currently, TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper requires that the user who runs activities is the same user who has done the synchronization and created the tables.</p> <p>See Synchronizing SmartMapper ER Model regarding the Synchronization.</p>

Caching Tab

The **Caching** tab contains the following fields.

Field	Module Property?	Description
Enable Caching	No	Select to enable caching. Clear to disable caching.
Cache Type	No	<p>Select a caching type from the list. Two types are available:</p> <ul style="list-style-type: none"> • LOCAL: stores the caching data to a local memory. • ACTIVESPACES: stores the caching data using TIBCO ActiveSpaces. <div>  <p>To use TIBCO ActiveSpaces caching, ensure TIBCO ActiveSpaces is installed. Besides, add corresponding ActiveSpaces libraries to your system environment variables, see Adding ActiveSpaces Library into Environment Variables for more details.</p> </div>
The following fields are available when Local is selected in the Cache Type field.		
Maximum Number Of Entries	Yes	<p>The maximum number of entity instances to be stored in the cache.</p> <p>If the number of entity instances exceeds the specified number, the excess cache entries are culled from the cache using a least-recently-used algorithm.</p> <div>  <p>The value must be a positive integer, and it is not accurately equal to the numbers of the returned data generally.</p> </div>

Field	Module Property?	Description
Cache Synchronization Interval (sec)	Yes	<p>The time interval, in seconds, that cache housekeeping is performed.</p>  The value must be a positive integer.
The following fields are available when ActiveSpaces is selected in the Cache Type field.		
Metaspace Name	Yes	Specify the name of the metaspace to connect to.
Discovery URL	Yes	<p>Specify the discovery URL to use to discover each metaspace.</p>  Do not use the Rendezvous protocol in the discovery URL.
Listen URL	Yes	Specify the URL to use to listen for the incoming connections from new members to the metaspace.
Relationship Capacity	Yes	<p>Specify the maximum number of relationship instances per seeder for the space.</p> <p>If the number of relationship instances exceeds the specified number, the relationship instances that are not frequently used are deleted.</p>  When the value is -1, the capacity is set to infinite.
Replication All	No	Specify whether to replicate all the relationship data or not. Selecting this check box means that all the relationship data are replicated on all the seeders.
Distribution Role	No	<p>Select the distribution role for a newly joined member from the following options:</p> <ul style="list-style-type: none"> • SEEDER: the new member plays an active role in maintaining the space by providing CPU and RAM. • LEECH: the new member plays a passive role which has access to space data but provides no resources.  If you want to use LEECH as distribution role, make sure there is an as-agent that already joined metaspace, or a "No Enough SEEDER" error occurs.
Export	No	Generates an XML file, which includes the ActiveSpaces parameters of all the entities and relationships in this ER Model.

Synchronizing SmartMapper ER Model

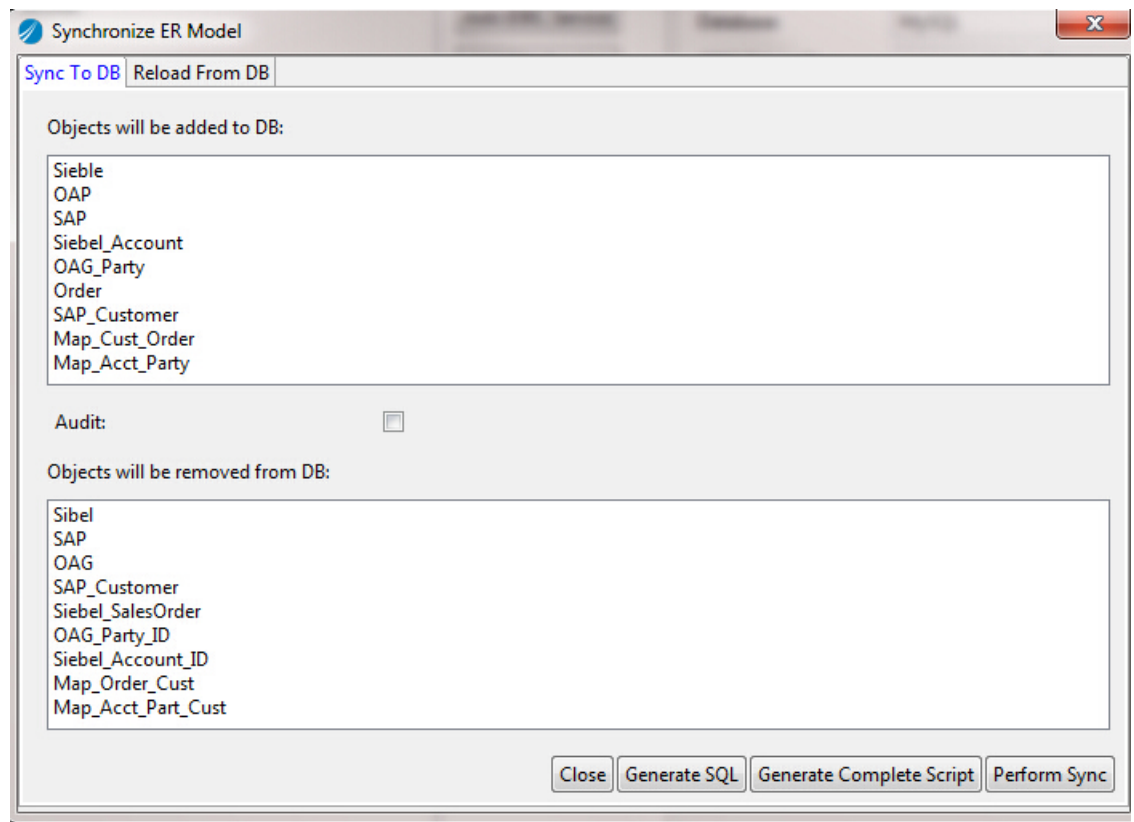
Synchronization creates the tables and sequences in a database that are required for storing a SmartMapper ER Model in the database.

Prerequisites

Before synchronizing SmartMapper ER Model, you are required to [create a SmartMapper ER Model shared resource](#) in TIBCO Business Studio.

Procedure

1. In TIBCO Business Studio, open the created SmartMapper ER Model.
2. In the Storages editor, click the added JDBC service.
3. Configure the JDBC settings properly.
See [JDBC-Based SmartMapper Service](#) for more details.
4. In the **JDBC Settings** tab, click **Sync ER Model w/DB**.
The **Synchronize ER Model** dialog is displayed, containing two configuration tabs:
 - [Sync To DB Tab](#)
 - [Reload From DB Tab](#)



5. Optional: In the **Sync To DB** tab, click **Audit** to enable auditing.
6. Use one of the following ways to perform the synchronization:
 - Click **Generate SQL** or **Generate Complete Script** to copy a SQL script and run the script in your database.

- Click **Perform Sync** to start the synchronization.
7. Close the dialog when the synchronization completes.


Sync To DB Tab

The **Sync To DB** tab contains the following fields.

Field	Module Property?	Description
Objects will be added to DB	No	The SmartMapper ER Model objects that are created in the database when synchronization is invoked.
Audit	No	<p>Select to enable auditing.</p> <ul style="list-style-type: none"> • For identity relationships, when auditing is enabled, a new table is created and an ADT prefix is added in the table name. • For association relationships, when auditing is enabled, the creating_date and modified_date attributes are added in the table.
Objects will be removed from DB	No	<p>The SmartMapper ER Model objects that are removed.</p> <p>The database tables for removed objects are not actually dropped during synchronization. Instead the tables are marked as deleted and are not used by TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper. That is, the data is left intact for removed objects.</p>
Objects will be changed into DB	No	The SmartMapper ER Model objects that are changed.
Close	No	Close this dialog.
Generate SQL	No	Generates a window that contains the SQL that is executed if you click Sync ER Model w/DB .
Generate Complete Script	No	Generates a complete SQL script for the current SmartMapper ER Model. Click Copy to Clipboard to copy the SQL script to the Clipboard.
Perform Sync	No	Creates or removes the TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper objects in the database.

Reload From DB Tab

The **Reload From DB** tab contains the following fields.


Field	Module Property?	Description
Objects will be added to Studio	No	The SmartMapper ER Model objects which are added to the project from the database.
Remove all not in Database?	No	<p>Select if you want to remove all the objects from the existing SmartMapper ER Model in the project.</p> <div>  <p>It is better to leave this check box clear unless you want to explicitly remove the SmartMapper ER Model objects from your project.</p> </div>
Objects will be removed from DB if the above option is selected	No	The SmartMapper ER Model objects that are removed.
Close	No	Close this dialog.
Perform Reload	No	Imports SmartMapper ER Model objects from the database to the existing project.

Adding ActiveSpaces Library into Environment Variables

To use the TIBCO ActiveSpaces caching, you have to add the ActiveSpace library to the environment variables.

Add the ActiveSpaces library according to the platform that you are using:

- For Windows:
 - add `AS_HOME/version_number/lib` to the path environment variable.
- For Linux:
 - add `AS_HOME/version_number/lib` to the path environment variable.
 - add `LD_LIBRARY_PATH=AS_HOME/version_number/lib` to the environment variables.



To deploy the application to TIBCO Enterprise Administrator, add `AS_HOME` to the environment variables.
- For Mac OS:
 - add `AS_HOME/version_number/lib` to the path environment variable.
 - add `LD_LIBRARY_PATH=AS_HOME/version_number/lib` to the environment variables.
 - add `AS_HOME` to the environment variables.
 - copy jar files in the `AS_HOME/version_number/lib` directory to the `TIBCO_HOME/tibcojre64/1.8.0/lib` folder.

File-Based SmartMapper Service

The file-based service is used for small sets of data that are not often edited. Data stored in a file-based service are always read entirely into the memory. File-based service is generally used for development and demos or when the data is more or less static.






The [Bulk Load](#), [Bulk Extract](#), [Dynamic Lookup](#), and [Dynamic Delete](#) activities are not supported for use with the File-Based SmartMapper Service.

When using [Lookup](#) and [SmartMapper Wizard](#) activities, if you use entity as the input format, it is also not supported for use with the File-Based SmartMapper Service.

Configuration Tab

The **Configuration** tab contains the following fields.

Field	Module Property?	Description
Name	No	The name to be displayed as the label for the storage service.
Active	No	Select this check box to set this storage service as active. If you set many storage services, only one service can be active at a time.
Data File	Yes	<p>Specify the file that stores the cross-referencing data.</p> <p>Click  to select an existing data file, or click  to add a new file.</p> <p> Only XML format is the supported.</p>

SmartMapper Palette

TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper provides various activities to design an ActiveMatrix BusinessWorks process.

The SmartMapper Palette consists of the following activities:

- [Bulk Extract](#)
- [Bulk Load](#)
- [Create Entity Instance](#)
- [Update Entity Instance](#)
- [Delete Entity Instance](#)
- [Create Relationship Instance](#)
- [Add to Relationship Instance](#)
- [Delete Relationship Instance](#)
- [Lookup](#)
- [Dynamic Lookup](#)
- [Dynamic Delete](#)
- [Generate Key](#)
- [SmartMapper Wizard](#)

Bulk Extract

Use this activity to bulk extract cross-referencing data by providing an entity name or a relationship name.

Because bulk extraction results in a large number of records, you can retrieve results in smaller subsets.





The Bulk Extract activity is not supported for use with the File-Based SmartMapper Service.

Bulk Extract Activity - General Tab

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

Field	Module Property?	Description
Name	No	The name to be displayed as the label for the activity in the process.
Output Format	No	<p>The output structure (in the Output tab) changes depending on your selection. Select one output format from the following options:</p> <ul style="list-style-type: none"> • Entity Format: select to return entity only. • One Participant Relationship Format: select if the relationship is an identity relationship. That is, there is one or multiple participants per relationship. • Two Participant Relationship Format: select if the relationship is an association relationship. That is, there are only two participants in each relationship and both relationships are many-to-many.



Field	Module Property?	Description
ER Model	Yes	<p>A SmartMapper ER Model defines a set of relationships and their participating entities.</p> <p>Click the  icon to select a SmartMapper ER Model.</p>
Process in Subsets	No	<p>Select the check box to process the result set in smaller batches rather than process the entire result set at once.</p> <ul style="list-style-type: none"> When cleared, complete results are returned on one execution. If the result set is large, you might want to use a distributed queue to improve performance by balancing the load among services. When selected, performance cannot be improved significantly by using a distributed queue. <p>A <code>subsetSize</code> input element and a <code>lastSubset</code> output element are added.</p>
Show Record Timestamp	No	<p>Select the check box to include the timestamps in the report.</p> <div>  <p>To add the timestamp field when you synchronize SmartMapper ER Model, make sure that Audit is selected.</p> </div>



Bulk Extract Activity - Description Tab

Provide a short description for the activity.

Bulk Extract Activity - Input Tab

The **Input** tab contains a `root` node, which contains the following sub-nodes. The input schema is often changed depending on the **Output Format** field that is specified in the **General** tab.

Input Item	Data Type	Description
entityName	String	<p>The name of the entity to be extracted.</p> <div>  <p>It is displayed only when Entity Format is selected in the Output Format field in the General tab.</p> </div>
relationshipName	String	<p>The name of the relationship to be extracted.</p> <div>  <p>It is displayed only when a relationship format is selected in the Output Format field in the General tab.</p> </div>
extractionFilter	Complex	<p>The extraction filter to filter out the participants to be extracted.</p> <p>See Extraction Filter for more details about each element of the extraction filter.</p>



Input Item	Data Type	Description
subsetSize	Integer	<p>The number of records to process in each batch. You can process smaller batches of rows instead of retrieving one large result set.</p> <div>  It is displayed only when the Process In Subsets check box is selected in the General tab. </div> <div>  The value must be a positive integer. </div>

Extraction Filter

Use an extraction filter to filter the entities and relationships to be extracted.

Extraction Filter Component

Each extraction filter contains the following elements.

Input Item	Data Type	Description
entityFieldName	String	<p>The field name of the entity to be extracted. The name is mapped to the column name in the WHERE clause.</p> <div>  This element is displayed only when Entity Format is selected in the General tab. </div>
participantFieldName	String	<p>The name of the relationship, which consists of the participant name and the field name of the participant separated by dot. The name is mapped to the column name of the WHERE clause.</p> <p>createTimestamp and updateTimestamp are also supported for timestamp-based filter criteria.</p> <div>  This element is displayed only when relationship format is selected in the General tab. </div>
where	String	<p>The where element is mapped to the operator of the WHERE clause and the value is mapped to the value of the WHERE clause.</p> <p>The operator is an SQL operator such as ">", "<", "=", or "like".</p>
orWhere	String	<p>The orWhere element is mapped to the operator of the WHERE clause and the value is mapped to the value of the WHERE clause.</p> <p>The operator is an SQL operator such as ">", "<", "=", or "like".</p>



To use the filters, an operator is required for the values of the where and orWhere elements.

Multiple Filters

Each instance of the extraction filter is mapped to a condition in the SQL WHERE clause (WHERE <column> <operator> <value>).

When multiple extraction filter instances are specified for the same entity or relationship, each of them is AND'ed if the operator and value are specified using the `where` element. Conditions are OR'ed together if the operator and value are specified using the `orWhere` element.

For example, the following two filter instances:

- ExtractionFilter Instance 1: `partFieldName = 'id', where = '>345'`
- ExtractionFilter Instance 1: `partFieldName = 'id', orWhere = 'like%345%'`

are mapped to the following SQL WHERE Clause:

WHERE id > '345' OR id 'like%345%'



In a filter, the value of the `orWhere` takes effect only when the value for the `where` element is empty.

If the first filter uses the `orWhere` element, SmartMapper always treats it as `where` element .

Limitations

Limitations to the extraction filter are as follows:

- Filtering criteria cannot be parenthesized.
- When adding two filters and each filter is a different participant, the filters are only combined as OR; and if one filter is a participant and the other filter is a `createTimestamp`, the two filters are also only combined as OR.
- When adding filters for timestamps, the filter names must only be `createTimestamp` or `updateTimestamp`.


Bulk Extract Activity - Output Tab

The output schema depends on the output format that is specified in the **General** tab.

Entity Format


The following table lists the output items for the activity, when **Entity Format** is selected from the **Output Format** field in the **General** tab.

Output Item	Data Type	Description
name	String (repeating)	The name of the entity selected from Output Format field in the General tab.
createTimestamp	Date	The date when the timestamp for a record is created.
updateTimestamp	Date	The date when the timestamp for a record is updated.

Output Item	Data Type	Description
field	Complex	<p>This element contains both the key and non-key fields. The field order is dictated by the entity's schema definition.</p> <ul style="list-style-type: none"> • index (integer) • name (string) • value (string)
lastSubset	Boolean	<p>A value of true indicates that the current result set is the last subset.</p> <div>  <p>This element is displayed only when the Process in Subsets check box is selected.</p> </div>


One Participant Relationship Format

The following table lists the output items for the activity, when **One Participant Relationship Format** is selected from the **Output Format** field in the **General** tab.

Output Item	Data Type	Description
name	String (repeating)	The name of the relationship selected from Output Format field in the General tab.
relationshipInstance	None	<p>This element contains the following subordinate elements:</p> <ul style="list-style-type: none"> • createTimestamp: the date when the timestamp for a record is created. • updateTimestamp: the date when the timestamp for a record is updated. • participant: this element contains the following sub-elements: <ul style="list-style-type: none"> – name (string): the name of the participant. – createTimestamp: the date when the timestamp for a record is created. – updateTimestamp: the date when the timestamp for a record is updated. – field: contains both the key and non-key fields. The field order is dictated by the entity's schema definition.
lastSubset	Boolean	<p>A value of true indicates that the current result set is the last subset.</p> <div>  <p>This element is displayed only when the Process in Subsets check box is selected.</p> </div>

Two Participant Relationship Format

The following table lists the output items for the activity, when **Two Participant Relationship Format** is selected from the **Output Format** field in the **General** tab.

Output Item	Data Type	Description
name	String (repeating)	The name of the relationship selected from Output Format field in the General tab.
relationshipInstance	None	<p>This element contains the following subordinate elements:</p> <ul style="list-style-type: none"> • createTimestamp: the date when the timestamp for a record is created. • updateTimestamp: the date when the timestamp for a record is updated • participant1: this element contains the following sub-elements: <ul style="list-style-type: none"> – name (string): the name of the participant. – createTimestamp: the date when the timestamp for a record is created. – updateTimestamp: the date when the timestamp for a record is updated. – field: contains both the key and non-key fields. The field order is dictated by the entity's schema definition. • participant2: this element contains the following sub-elements: <ul style="list-style-type: none"> – name (string): the name of the participant. – createTimestamp: the date when the timestamp for a record is created. – updateTimestamp: the date when the timestamp for a record is updated. – field: contains both the key and non-key fields. The field order is dictated by the entity's schema definition.
lastSubset	Boolean	<p>A value of true indicates that the current result set is the last subset.</p> <div>  <p>This element is displayed only when the Process in Subsets check box is selected.</p> </div>

Bulk Extract Activity - Fault Tab

The **Fault** tab lists exceptions that are thrown by the activity.

Error Schema Element	Data Type	Description
errorCode	String	The error code for errors that are generated by the underlying SmartMapper engine.
msg	String	The error message.
input	String	A copy of the data that is input to the activity.

Bulk Load


Use this activity to bulk load cross-referencing data by providing an entity name or a relationship name. Unlike other activities, the Bulk Load activity does not stop if an error occurs. That is, if a record fails to load into the database, the activity will not stop.



The Bulk Load activity is not supported for use with the File-Based SmartMapper Service.

Bulk Load Activity - General Tab

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

Field	Module Property?	Description
Name	No	The name to be displayed as the label for the activity in the process definition.
Input Format	No	Choose an input format: <ul style="list-style-type: none"> • Entity Format: select to return entity only. • One Participant Relationship Format: select if the relationship is an identity relationship. That is, there is one or multiple participants per relationship. • Two Participant Relationship Format: select if the relationship is an association relationship. That is, there are only two participants in each relationship and both relationships are many-to-many.
ER Model	Yes	A SmartMapper ER Model defines a set of relationships and their participating entities. Click the  icon to select a SmartMapper ER Model.

Field	Module Property?	Description
Criteria	No	<p>Select one criterion to load entities:</p> <ul style="list-style-type: none"> • Name: for each entity, load records by name. • Index: index is the order in which the fields are defined in the schema. Index starts from 1. If specified, loading is by index.
Mode	No	<p>This field is only displayed when the Input Format field is set to One Participant Relationship Format. Select a loading mode:</p> <ul style="list-style-type: none"> • Create: any data load has a chance of collisions. When running in Create mode, the data is loaded in with minimal collision detection. This is the fastest, most efficient way to seed an empty store. See Create Mode Setting for details. • Add: this mode checks first for duplicate records and ignores them, rather than throwing an error. See Add Mode Setting for details. • Update: used to replace a relationship instance with a new one. See Update Mode Setting for details.
Enable Verbose Reporting	No	<p>When selected, a succeeded field and a failed field are reported for each record.</p> <p>When cleared, the report includes details only about records that caused errors. A concise summary of successful loads and loads with given warnings is generated.</p> <p>The report is displayed as part of the Output tab.</p>

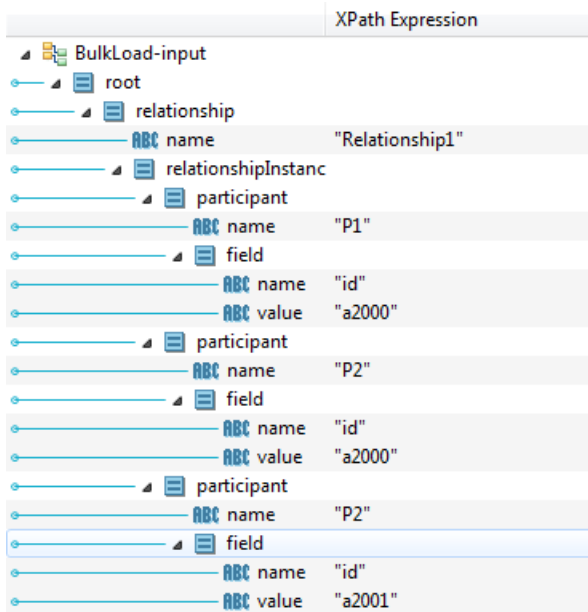
Create Mode Setting

If the Mode setting is **Create**, a participant with the **Cardinality is many** option selected can be assigned with multiple values in the same relationship instance.

When **Create** is selected, all participants must be provided in a single relationship instance.

For example, there are two participants, P1 and P2. P1 has the **Cardinality is many** option cleared (it has a one-to-one relationship) and P2 has the **Cardinality is many** option selected (it has a many-to-many relationship). P2 has two values assigned in the same relationship instance.

The following figure shows the input of the Bulk Load activity where the Input format is set to **One Participant Relationship Format** and the **Mode** setting is set to **Create**.



Add Mode Setting

If the Mode setting is **Add**, a participant with the **Cardinality is many** option selected cannot be assigned multiple values in the same relationship instance. Instead, each value must be assigned using a unique relationship instance.

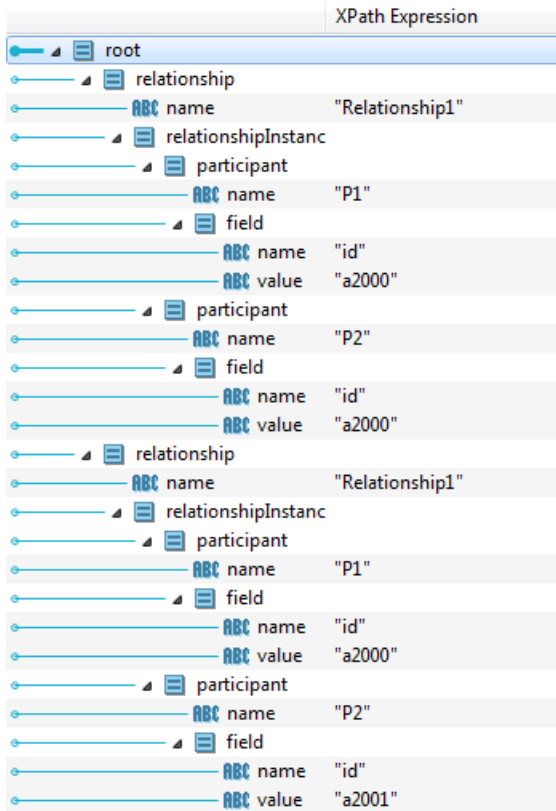
When this mode is selected, only one participant is primary. That is, participant entries can be split across multiple relationship instances, but each relationship instance must contain one participant (primary participant) that is already included in some other relationship instance.

For examples, there are two participants, P1 and P2. P1 has the **Cardinality is many** option cleared (it has a one-to-one relationship) and P2 has the **Cardinality is many** option selected (it has a many-to-many relationship). P2 has two values assigned and each value is set in a different relationship instance.



When there are multiple participants with one-to-one relationships defined, only one participant is displayed in subsequent relationship instances.

The following figure shows the input of the Bulk Load activity where the Input format is set to **One Participant Relationship Format** and the **Mode** setting is set to **Add**.



Update Mode Setting

If the Mode setting is **Update**, lookup is performed based on the primary participant, and if no instance is found, a new one is created. If a lookup returns an existing relationship instance, it is replaced with the new participant.



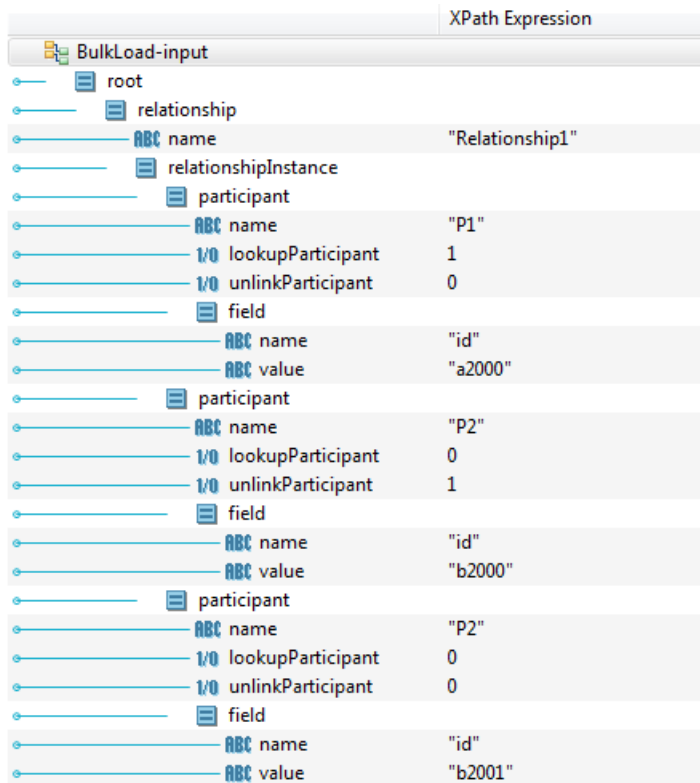
Only one participant can be `lookupParticipant`. The other participant can be either `lookupParticipant` or `unlinkParticipant`.

There are two participants, P1 and P2. The `lookupParticipant` value of the P1 participant is 1, which means that the lookup is performed for a relationship instance based on this participant.

If there is a P2 `unlinkParticipant` (value 1), then the given value in its field (b200) is replaced with the value given for the Replacement Participant (b2001).

If there is no P2 `unlinkParticipant` (value 0), then the existing values in the database (such as b2003, b2004, b2005) are replaced with the value given for the Replacement Participant (b2001). When there are no values to replace, a new value (b2001) is added.

The following figure shows the input of the Bulk Load activity where the Input format is set to **One Participant Relationship Format** and the **Mode** is set to **Update**.



Bulk Load Activity - Description Tab

Provide a short description for this activity.

Bulk Load Activity - Input Tab

The input schema depends on the input format that is specified in the **General** tab.


Entity Format

The following table lists the input items for the activity, when **Entity Format** is selected from the **Input Format** field in the **General** tab.

Input Item	Data Type	Description
name	String (repeating)	The name of the entity selected from the Input Format field in the General tab.
field	Complex	<p>This element contains both the key and non-key fields. The field order is dictated by the entity's schema definition.</p> <ul style="list-style-type: none"> index (integer) name (string) value (string)

One Participant Relationship Format

The following table lists the input items for the activity, when **One Participant Relationship Format** is selected from the **Input Format** field in the **General** tab.

Input Item	Data Type	Description
name	String (repeating)	The name of the relationship selected from the Input Format field in the General tab.
relationshipInstance	None	<p>For the relationship instance, provide:</p> <ul style="list-style-type: none"> name (string): the name of the participant. field: contains both the key and non-key fields. The field order is dictated by the entity's schema definition. <ul style="list-style-type: none"> index (integer) name (string) value (string) <div>  <p>The following two nodes are provided if Update is selected in the Mode field in General tab.</p> </div> <ul style="list-style-type: none"> lookupParticipant: looks for a relationship instance based on this participant. Value 1 or 0 indicates whether the lookup is performed. unlinkParticipant: updates the existing participant with the new value. Value 1 or 0 indicates whether the update is performed. <p>See the Mode field in the General tab for more details.</p>

Two Participant Relationship Format

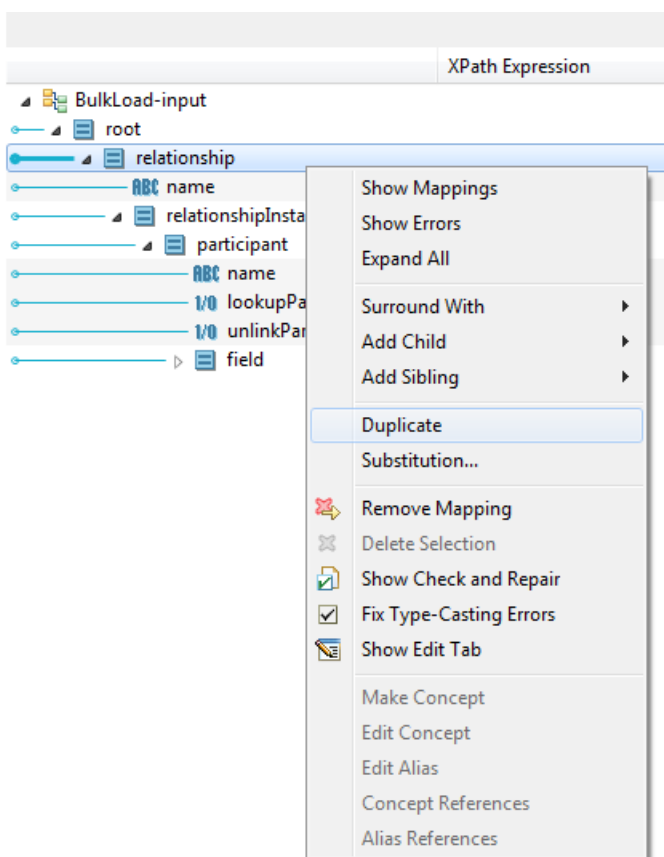
The following table lists the input items for the activity, when **Two Participant Relationship Format** is selected from the **Input Format** field in the **General** tab.

Input Item	Data Type	Description
name	String (repeating)	The name of the relationship selected from the Input Format field in the General tab.

Input Item	Data Type	Description
relationshipsInstance	None	<p>For each relationship instance, provide:</p> <ul style="list-style-type: none"> • participant1: this element contains the following sub-elements: <ul style="list-style-type: none"> – name(string) : the name of the participant. – field: contains both the key and non-key fields. The field order is dictated by the entity's schema definition. • participant2: this element contains the following sub-elements: <ul style="list-style-type: none"> – name (string): the name of the participant. – field: contains both the key and non-key fields. The field order is dictated by the entity's schema definition.

By right-clicking the node and selecting **Duplicate**, you can duplicate the following input nodes:

- entity
- field
- relationship
- participant



Bulk Load Activity - Output Tab

The display in the **Output** tab changes depending on the **Input Format** field setting as specified under the **General** tab.

If **Enable Verbose Reporting** is selected in the **General** tab, the structure in the **Output** tab includes a detailed report with an entry with regard to each input record.

If **Enable Verbose Reporting** is cleared, the structure in the **Output** tab only includes the number of succeeded loads in the integer node `LoadStatusReport`.

Bulk Load Activity - Fault Tab

The **Fault** tab lists exceptions that are thrown by the activity.


Error Schema Element	Data Type	Description
errorCode	String	The error code for errors that are generated by the underlying SmartMapper engine.
msg	String	The error message.
input	String	A copy of the data that was input to the activity.

Create Entity Instance

Use this activity to create the data for an entity instance without relating the instance to any other object.

Create Entity Instance Activity - General Tab

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

Field	Module Property?	Description
Name	No	The name to be displayed as the label for the activity in the process definition.
Entity	Yes	Click the  icon to select the entity to create an instance for.

Create Entity Instance Activity - Description Tab

Provide a short description for this activity.

Create Entity Instance Activity - Input Tab

The following table lists the input items for the activity.

Input Item	Data Type	Description
root	Complex	The schema for the entity to create. This is a node allowing for the creation of multiple entity instances.

Create Entity Instance Activity - Output Tab

The following table lists the output items for the activity.

Output Item	Data Type	Description
root	Complex	The schema for the entity to create. This is a repeating node allowing for the creation of multiple entity instances.

Create Entity Instance Activity - Fault Tab

The **Fault** tab lists exceptions that are thrown by the activity.


Error Schema Element	Data Type	Description
msgCode	String	The error code returned by the plug-in.
msg	String	The error message.
errCode	String	The error code for errors that are generated by the underlying SmartMapper engine.
root	String	The schemas that are same as those in the input items.

Update Entity Instance

Use this activity to update entity instance data. This affects all relationships that the entity instance participates in.

Update Entity Instance Activity - General Tab

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

Field	Module Property?	Description
Name	No	The name to be displayed as the label for the activity in the process definition.
Entity	Yes	Click the  icon to select the entity to update an instance of.

Update Entity Instance Activity - Description Tab

Provide a short description for this activity.

Update Entity Instance Activity - Input Tab

The following table lists the input items for the activity.

Input Item	Data Type	Description
oldObject	Complex	The schema for the entity. Map values here to identify the entity instance that is being updated.
newObject	Complex	Map values here to supply the new key values for this entity instance.

Update Entity Instance Activity - Output Tab

The following table lists the output items for the activity.

Output Item	Data Type	Description
oldObject	Complex	The schema for the entity. Map values here to identify the entity instance that is being updated.
newObject	Complex	Map values here to supply the new key values for this entity instance.

Update Entity Instance Activity - Fault Tab

The **Fault** tab lists exceptions that are thrown by the activity.


Error Schema Element	Data Type	Description
msgCode	String	The error code returned by the plug-in.
msg	String	The error message.
errCode	String	The error code for errors that are generated by the underlying SmartMapper engine.
root	String	The schemas that are same as those in the input items.

Delete Entity Instance

Use this activity to delete an orphaned entity instance from the system, that is, an entity instance which is not part of any relationship instance. An exception is thrown if the entity instance is part of a relationship instance.

Delete Entity Instance Activity - General Tab

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

Field	Module Property?	Description
Name	No	The name to be displayed as the label for the activity in the process definition.
Entity	Yes	Click the  icon to select the entity to delete an instance.

Delete Entity Instance Activity - Description Tab

Provide a short description for this activity.

Delete Entity Instance Activity - Input Tab

The **Input** tab contains a **root** node, which includes the following nodes.

Input Item	Data Type	Description
root	Complex	The key schemas for the entity to delete. This is a node allowing for the removal of multiple entity instances.

Delete Entity Instance Activity - Output Tab

The following table lists the output items for the activity.

Output Item	Data Type	Description
root	Complex	The schema for the entity to delete. This is a repeating node allowing for the removal of multiple entity instances.

Delete Entity Instance Activity - Fault Tab

The **Fault** tab lists exceptions that are thrown by the activity.

Error Schema Element	Data Type	Description
msgCode	String	The error code returned by the plug-in.
msg	String	The error message.



Error Schema Element	Data Type	Description
errCode	String	The error code for errors that are generated by the underlying SmartMapper engine.
root	String	The schemas that are same as those in the input items.

Create Relationship Instance

Use this activity to create a set of entity instances and associate them in a relationship instance. The activity also seeds data into its respective entity table, if the data is not there.

Create Relationship Instance Activity - General Tab

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

Field	Module Property ?	Description
Name	No	The name to be displayed as the label for the activity in the process definition.
Relationship	Yes	<p>The relationship for which this relationship instance is created.</p> <p> When a selected relationship contains TIBCO-Managed entities, this activity generates keys for them if no values are mapped to them.</p>
ShowNonKeys	No	<p>When selected, non-key attributes are displayed.</p> <p> This check box is selected by default.</p>

Create Relationship Instance Activity - Description Tab

Provide a short description for this activity.

Create Relationship Instance Activity - Input Tab

The input contains the keys of all the entities that participate in the chosen relationship. Map values into the keys to create the entity instances and associate them in a new relationship instance.

Create Relationship Instance Activity - Output Tab

The output contains the data for all the entity instances that were created. This is a copy of the input data.

Create Relationship Instance Activity - Fault Tab

The **Fault** tab lists exceptions that are thrown by the activity.

Error Schema Element	Data Type	Description
msgCode	String	The error code returned by the plug-in.

Error Schema Element	Data Type	Description
msg	String	The error message.
errCode	String	The error code for errors that are generated by the underlying SmartMapper engine.
keys	String	The schemas that are same as those in the input items.

Add to Relationship Instance



Use this activity to add entity instances to an existing relationship instance.



This activity only supports Identity relationships.

Add to Relationship Instance Activity - General Tab

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

Field	Module Property?	Description
Name	No	The name to be displayed as the label for the activity in the process definition.
Participant	Yes	<p>The participant for an existing entity instance that is already a member of a relationship instance.</p> <p>This entity instance is used to locate the relationship instance to which the new entity instances are added.</p> <div>  <p>When the selected participant is a TIBCO-Managed entity, this activity generates key for the participant if no values are mapped to it.</p> </div>
ShowNonKeys	No	<p>When selected, non-key attributes are displayed.</p> <div>  <p>This check box is selected by default.</p> </div>


Add to Relationship Instance Activity - Description Tab

Provide a short description for this activity.

Add to Relationship Instance Activity - Input Tab

The **Input** tab contains a **root** node, which includes the following nodes.

Input Item	Data Type	Description
baseEntry	Complex	The schema for the existing entity instance specified in the Participant field. Map values for an entity instance that already exists and is part of an existing relationship instance.

Input Item	Data Type	Description
newEntries	Complex	Contains the schemas for the other participants in this relationship.  Make sure the cardinalities of the participants are many in this input.

Add to Relationship Instance Activity - Output Tab

The following table lists the output items for the activity.

Output Item	Data Type	Description
baseEntry	Complex	The schema for the existing entity instance specified in the Participant field. Map values for an entity instance that already exists and is part of an existing relationship instance.
newEntries	Complex	Contains the schemas for the other participants in this relationship.

Add to Relationship Instance Activity - Fault Tab

The **Fault** tab lists exceptions that are thrown by the activity.

Error Schema Element	Data Type	Description
msgCode	String	The error code returned by the plug-in.
msg	String	The error message.
errCode	String	The error code for errors that are generated by the underlying SmartMapper engine.
root	String	The schemas that are same as those in the input items.

Delete Relationship Instance

Use this activity to delete relationship instances.



This activity does not delete entity instances themselves, rather it removes the user specified memberships if they exist. This activity occasionally leaves orphan data, such as side entries in an identity relationship instance that is left; in this case, you have to use this activity multiple times to clean the data.

Delete Relationship Instance Activity - General Tab

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

Field	Module Property?	Description
Name	No	The name to be displayed as the label for the activity in the process definition.

Field	Module Property?	Description
Relationship	Yes	The relationship from which to remove relationship instances.

Delete Relationship Instance Activity - Description Tab

Provide a short description for this activity.

Delete Relationship Instance Activity - Input Tab

The **Input** tab contains a `root` node, which includes the following nodes.

Input Item	Data Type	Description
keys	Complex	Contains the key schemas for the participants in this relationship. Map values here for each of the entity instances that are removed from this relationship.

Delete Relationship Instance Activity - Output Tab

The following table lists the output items for the activity.

Output Item	Data Type	Description
keys	Complex	Contains the key schemas for the participants in this relationship. Map values here for each of the entity instances that are removed from this relationship.

Delete Relationship Instance Activity - Fault Tab

The **Fault** tab lists exceptions that are thrown by the activity.


Error Schema Element	Data Type	Description
msgCode	String	The error code returned by the plug-in.
msg	String	The error message.
errCode	String	The error code for errors that are generated by the underlying SmartMapper engine.
keys	String	The schemas that are same as those in the input items.

Lookup

Use this activity to accept a participant instance as input, and then retrieve one or more participant instances by performing lookups on entity instances and relationship instances.

Lookup Activity - General Tab

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

Field	Module Property?	Description
Name	No	The name to be displayed as the label for the activity in the process definition.
Input Format	No	The type of data to be looked up. Two data types are supported: Entity and Relationship .
Entity	Yes	<p>The entity that you want to look up.</p> <p>Note: This field is only available when Entity is selected in the Input Format field.</p> <div>  <p>Entity lookup does not support for use with File-Based SmartMapper Service.</p> </div>
Input Participant	Yes	<p>The participant is used to locate a relationship instance.</p> <p>Note: This field is only available when Relationship is selected in the Input Format field.</p>
Output Participant	Yes	<p>The list of participants to look up. The lookup activity supports lookup of only one output participant.</p> <p>Note: This field is only available when Relationship is selected in the Input Format field.</p>

Lookup Activity - Description Tab

Provide a short description for this activity.

Lookup Activity - Input Tab

The following table lists the input items for the activity.

Input Item	Data Type	Description
Entity_name	Complex	The schema for the input participant. These values are used to find relationship instances and entity instances.

Lookup Activity - Output Tab

The following table lists the output items for the activity.

Output Item	Data Type	Description
root	Complex	<p>The schemas for the participants that are found by the lookup. These are the entity instances or relationship instances that are associated with the input participants.</p> <p>For entity instance lookup, the output data contain nodes only for the output participants that are found. Output participants that are not found return no nodes in the output data.</p> <p>For relationship instance lookup, the output data contain both input and output participants. And output participants that are not found return input node in the output data.</p>

Lookup Activity - Fault Tab

The **Fault** tab lists exceptions that are thrown by the activity.

Error Schema Element	Data Type	Description
msgCode	String	The error code returned by the plug-in.
msg	String	The error message.
errCode	String	The error code for errors that are generated by the underlying SmartMapper engine.
Entity_name	String	The schemas that are same as those in the input items.

Dynamic Lookup

Use this activity to retrieve one or more participants by performing lookups on entity instances and relationship instances.


In a dynamic lookup, relationship and participant information is derived from the data, rather than modeled at design time. You can specify information during runtime (dynamically). You need not define data at design time. If a lookup fails, the output is empty. The activity also operates on a set of data, like the SmartMapper Wizard activity.



The Dynamic Lookup activity is not supported for use with the File-Based SmartMapper Service.

Dynamic Lookup Activity - General Tab

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


Field	Module Property?	Description
Name	No	The name to be displayed as the label for the activity in the process definition.
Input Format	No	Select the input format: Relationship or Entity . Dynamic Lookup is either key based or non-key based.
ER Model	Yes	A SmartMapper ER Model defines a set of relationships and their participating entities. Click the  icon to select a SmartMapper ER Model.
Criteria	No	Select one criterion to load entities: <ul style="list-style-type: none"> • Name: for each entity, lookup records by name. • Index: index is the order in which the fields are defined in the schema. Index starts from 1. If specified, loading is by index.
Show Record Timestamp	No	Select the check box to include the timestamps in the output.


Dynamic Lookup Activity - Description Tab

Provide a short description for this activity.

Dynamic Lookup Activity - Input Tab

The **Input** tab contains a **root** node, which includes the following nodes.

Input Item	Data Type	Description
relationship	Complex	<p>Provide the values as strings in the following items:</p> <ul style="list-style-type: none"> • name (string): name of the relationship • inputParticipant: name and field. This field is a complex structure (repeating). It includes only key fields. Field is either a name or an index depending on the value selected for Criteria. • outputParticipant: name. This name field could be empty, in which case all participants related to the given input participant are outputted. <div>  <p>This item is displayed only when Relationship is selected from the Input Format field in the General tab.</p> </div>

Input Item	Data Type	Description
entity	Complex	<p>Provide the values as strings in the following items:</p> <ul style="list-style-type: none"> • name (string): name of the entity • field (complex): contains both the key and non-key fields <ul style="list-style-type: none"> – index (integer) – name (string) – value (string) <p> This item is displayed only when Entity is selected from the Input Format field in the General tab.</p>

Dynamic Lookup Activity - Output Tab

This tab contains a copy of the input values with the output filled in.

Dynamic Lookup Activity - Fault Tab

The **Fault** tab lists exceptions that are thrown by the activity.

Error Schema Element	Data Type	Description
errorCode	String	The error code for errors that are generated by the underlying SmartMapper engine.
msg	String	The error message.
input	String	A copy of the data that was input to the activity.

Dynamic Delete

Use this activity to delete entity instances and relationship instances using dynamic runtime inputs. This activity is a smart deletion activity which means that it does not leave orphan data.




The Dynamic Delete activity is not supported for use with the file-based SmartMapper service.

Dynamic Delete Activity - General Tab

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

Field	Module Property?	Description
Name	No	The name to be displayed as the label for the activity in the process definition.



Field	Module Property?	Description
Input Format	No	<p>Select one of the two input formats:</p> <ul style="list-style-type: none"> • Relationship: a relationship is a collection of entities that you want to bind together in a particular way. • Entity: something that is being referenced, such as a customer record.
ER Model	Yes	<p>A SmartMapper ER Model defines a set of relationships and their participating entities.</p> <p>Click the  icon to select a SmartMapper ER Model.</p>
Criteria	No	<p>Select one criterion to load entities:</p> <ul style="list-style-type: none"> • Name: for each entity, lookup records by name. • Index: index is the order in which the fields are defined in the schema. Index starts from 1. If specified, loading is by index.

Dynamic Delete Activity - Description Tab

Provide a short description for this activity.



Dynamic Delete Activity - Input Tab

The **Input** tab contains a **root** node, which includes the following nodes.

Input Item	Data Type	Description
relationship ip	Complex	<p>Provide the values as strings in the following items:</p> <ul style="list-style-type: none"> • name (string): name of the relationship • inputParticipant (complex): name and field The field is a complex structure (repeating), which includes only key fields. It is either a name or an index, depending on the value selected for Criteria. • deletionParticipant (complex): name and field The field is a complex structure (repeating), which includes only key fields. It is either a name or an index, depending on the value selected for Criteria. <ul style="list-style-type: none"> – This entry could be empty, which means that the whole relationship instance is deleted. – If only the deletionParticipant name is specified, it means that only the entries belonging to the given participant are deleted. – If all of the deletionParticipant entry is specified, it means that only the specified participant is unlinked. – If the entered participant's cardinality is one and only one, it deletes the whole relationship instance; otherwise it will only delete that one side participant instance. <div>  <ul style="list-style-type: none"> • This item is displayed only when Relationship is selected from the Input Format field in the General tab. • When the relationship type is Association, the value of inputParticipant and deletionParticipant can not be the same. </div>
entity	Complex	<p>Provide the values as strings in the following items:</p> <ul style="list-style-type: none"> • name (string): name of the entity • field (complex): contains only key fields <ul style="list-style-type: none"> – index (integer) – name (string) – value (string) <div>  <p>This item is displayed only when Entity is selected from the Input Format field in the General tab.</p> </div>

Dynamic Delete Activity - Output Tab

The following table lists the output items for the activity.

Output Item	Data Type	Description
relationship	Complex	<p>Contains the following elements:</p> <ul style="list-style-type: none"> • <code>StatusCode</code> (string) • <code>Msg</code> (string) • <code>name</code> (string): name of the relationship • <code>instanceID</code> (string) • <code>inputParticipant</code> (complex): this structure contains the relationship name and field (with name and value strings) • <code>deletedParticipant</code> (complex): this structure contains the relationship name and field (with name and value strings) <p> This item is displayed only when Relationship is selected from the Input Format field in the General tab.</p>
entity	Complex	<p>Contains the following elements:</p> <ul style="list-style-type: none"> • <code>StatusCode</code> (string) • <code>Msg</code> (string) • <code>name</code> (string): name of the entity • <code>field</code> (complex): this structure contains the index, entity field name and entity field value. <p> This item is displayed only when Entity is selected from the Input Format field in the General tab.</p>

Dynamic Delete Activity - Fault Tab

The **Fault** tab lists exceptions that are thrown by the activity.

Error Schema Element	Data Type	Description
errorCode	String	The error code returned by the plug-in.
msg	String	The error message.

Generate Key

Use this activity to generate a unique key for an entity. The operation does not create an entity but it takes the entity as a parameter to generate the key.

Generate Key Activity - General Tab

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

Field	Module Property?	Description
Name	No	The name to be displayed as the label for the activity in the process definition.
Entity	Yes	The entity for which a key is being generated.

Generate Key Activity - Description Tab

Provide a short description for this activity.

Generate Key Activity - Output Tab

The following table lists the output items for the activity.

Output Item	Data Type	Description
id	String	The generated key value.

Generate Key Activity - Fault Tab

The **Fault** tab lists exceptions that are thrown by the activity.

Error Schema Element	Data Type	Description
msgCode	String	The error code returned by the plug-in.
msg	String	The error message.
errCode	String	The error code for errors that are generated by the underlying SmartMapper engine.

SmartMapper Wizard

Use this activity to do lookups and maintain mappings within a single activity.

The SmartMapper Wizard activity sends requests to SmartMapper services in batches. The SmartMapper Wizard activity improves the performance and has different implications for the different service types.

- For the JDBC-Based SmartMapper Service, all operations invoked by the SmartMapper Wizard activity are done in one transaction.



- For the File-Based SmartMapper Service, all operations invoked by the SmartMapper Wizard activity are saved with one write to the file. For example, if the SmartMapper Wizard activity involves creating three relationship instances, those three edits of the data result in one write operation to the file.



Entity lookup does not support for use with File-Based SmartMapper Service.

SmartMapper Wizard Activity - General Tab

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

Field	Module Property?	Description
Name	No	The name to be displayed as the label for the activity in the process definition.
ER Model	Yes	<p>An ER Model defines a set of relationships and their participating entities.</p> <p>Click the  icon to select a SmartMapper ER Model.</p>
Verbose	No	<p>If selected, and the Error on empty results field in the SmartMapper Calls tab is cleared, the Wizard generates the following output:</p> <ul style="list-style-type: none"> groups — existing output. SmartMapper Exception. path. Path (XPath) pointing to the input member for which the Lookup failed. errorCode. Error code. msg. Error message, detailing cause of error and possible resolution . inputRow. Row of input data (entity or relationship) that caused this error. This could be used as is for retries later on. <div>  <p>This check box does not take effects when Maintain Mapping - One to One (xref_map) or Maintain Mapping - One to Many (xref_map) is selected from Invoke SmartMapper list in the SmartMapper Calls tab.</p> </div>

SmartMapper Wizard Activity - Description Tab

Provide a short description for this activity.

SmartMapper Wizard Activity - SmartMapper Calls Tab

The **SmartMapper Calls** tab annotates the input schema for logical locations for keys or references.



The fields in this tab are available only when you have created an input schema in the [Input Editor](#) tab.

In **SmartMapper Calls** tab, the selected field is not required to be part of the key. The location is the path where the mapper establishes context.

The **Input Participant** and **Output Participant** fields are for participants available in the **Relationships** folder.



All operations within a SmartMapper Wizard activity must use the same SmartMapper ER Model.

The **SmartMapper Calls** tab contains the following fields.

Field	Data Type	Description
Input Schema	Complex	The schema tree in the left pane. It is used to browse the input schema and select the nodes where a SmartMapper operation is invoked.
Invoke SmartMapper	Complex	<p>Select from the following operations to invoke a SmartMapper operation for the selected schema.</p> <ul style="list-style-type: none"> • Lookup Relationship (xref_lookup) • Lookup Entity (xref_lookup) • Maintain Mapping - One to One (xref_map) This operation does not apply to the association relationship. • Maintain Mapping - One to Many (xref_map) This operation does not apply to the association relationship. <p>See Invoke SmartMapper Operations for more information.</p>

Invoke SmartMapper Operations

Use the SmartMapper Wizard activity to invoke a batch of relationship lookup operations, entity lookup operations, and mapping maintenance operations.

Lookup Relationship (xref_lookup)

This operation takes a participant input and returns an output participant. The two participants are the parameters. Multiple instances are returned if the output participant is in the many relationship.


If the input evaluates to null, a lookup is not performed. Such a lookup returns null unless you select the check box to throw an exception.

Field	Description
Input Participant	<p>The participant used to look up.</p> <p> Same ER Model must be used throughout the activity.</p>

Field	Description
Output Participant	The participant that is being looked-up.
Error on Empty Result	When selected, the activity throws an exception if the lookup fails to find the output participant. If this check box is cleared, see the Verbose field description in the General tab.

Lookup Entity (xref_lookup)

With this operation, you can look up entities according to the input data.

Field	Description
Entity	Specify the entity is being looked up.  Same ER Model must be used throughout the activity.
Error on Empty Result	When selected, the activity throws an exception if the lookup fails to find the output participant. If this box is cleared, see the Verbose field description under the General tab.


Maintain Mapping - One to One (xref_map)

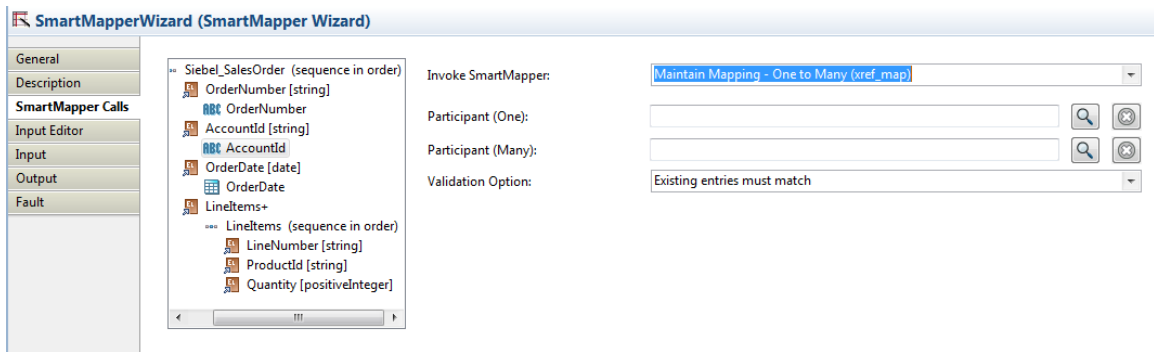
This operation maintains a one-to-one mapping between two participants. This operation creates and associates two entity instances within a relationship instance. The creation/association only occurs if the relationship instance does not already exist.

Field	Description
Participant	Select the two participants that are being maintained.

Maintain Mapping - One to Many (xref_map)

This operation maintains a one-to-many mapping. This operation employs the same semantics as the previous Maintain Mapping operation, but this operation also creates and associates entity instances in a one-to-many mapping.

Field	Description
Participant (One)	The participant on the one side or many side of the one-to-many association.
Participant (Many)	The participant on the many side of the one-to-many association. This participant must have selected the Cardinality Is Many check box in the ER Model definition.
Validation Option	<p>Choose a validation behavior when the relationship instance already exists:</p> <ul style="list-style-type: none"> • Existing entries must match: in this case, the values passed into the Input tab must match the existing values with no additions or removal of entries on the many side of the relationship. • Replace exiting entries: in this case, the values passed into the Input tab are meant to replace any existing values on the many side of the relationship. • Add any new entries: in this case, the values passed into the Input tab are to be added if they do not already exist in the many side of the relationship. <div>  <ul style="list-style-type: none"> • If the input value does not exist in one relationship instance and the selected option is not NONE, SmartMapper Wizard creates a new relationship instance. • If the input value exists in one relationship instance and the selected option is Add any new entries, SmartMapper Wizard creates nothing. • The participant of many side cannot be empty no matter if it is a TIBCO-Managed entity. </div>



SmartMapper Wizard Activity - Input Editor Tab

In the **Input Editor** tab, you can edit an input element by adding a complex anonymous type, complex element, primitive element, and so on.

See *TIBCO ActiveMatrix BusinessWorks Application Development* for more details about how to create an input schema.



Do not add an element of "Any Element".

SmartMapper Wizard Activity - Input Tab

The input schema depends on configurations in the **Input Editor** tab and **SmartMapper Calls** tab.

The schema includes nodes for each operation defined in the **SmartMapper Calls** tab. These nodes are placed in the locations specified in the **SmartMapper Calls** tab. These SmartMapper nodes represent the input data to the SmartMapper operations defined in the **SmartMapper Calls** tab.

Dragging the Schema to the input fields brings up the autofill wizard to set all non-SmartMapper fields.

Only use copy on sub-schemas that do not contain cross-reference calls, to ensure that your final object have both the lookups and the normal data in one object.

The SmartMapper Wizard allows calls to be batched to the database efficiently. Reads are occasionally done in parallel. Writes are occasionally done in one transaction. Exception semantics allow the capture of all lookup failures into one exception schema. Any real exception terminates the activity and rolls back any writes. All configurations are set up in the service.

Your next activity must be a Mapper.



In SmartMapper Wizard, the input field values are optional if both sides of the input entities participating in a relationship are null. Otherwise, you get a message that the key value is not present if one of the entities is null.

SmartMapper Wizard Activity - Output Tab

The output schema is based on the input schema. SmartMapper nodes have also been placed in the output schema.

You can view this in debug mode to see the new data.

SmartMapper Wizard Activity - Fault Tab

The **Fault** tab lists exceptions that are thrown by the activity.

Error Schema Element	Data Type	Description
WizardOperationException		

Error Schema Element	Data Type	Description
msgCode	String	The error code returned by the plug-in.
msg	String	The text that describes the error.
errCode	String	The error code for errors that are generated by the underlying SmartMapper engine.
operationType	String	The type of the operation that failed: LOOKUP_RELATIONSHIP, LOOKUP_ENTITY, MAINTAIN_ONE_TO_ONE, or MAINTAIN_ONE_TO_MANY.
path	String	The XPath path of the operation that failed.
input	String	A copy of the input data for the operation that failed.
SmartMapperException		
msgCode	String	The error code returned by the plug-in.
msg	String	The text that describes the error.
errCode	String	The error code for errors that are generated by the underlying SmartMapper engine.
ActivityTimedOutException		
msg	String	The text that describes the error.
msgCode	String	The error code returned by the plug-in.

SmartMapper Tools

TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper offers command-line tools. You can load and extract data in bulk, compare SmartMapper ER Model files, migrate relationship and entity data, and upgrade databases to correspond with different versions of TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper.

TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper offers the five following tools:

- [Bulk Extract Tool](#)
- [Bulk Load Tool](#)
- [Differencing Tool](#)
- [Migration Tool](#)
- [Database Upgrade Tool](#)

A database connection is required to run the command-line tools. Instead of specifying the database connection parameters one by one, you can prepare the databases before running the command-line tools. See [Configuring Database Connection](#) for more details.



Before using the SmartMapper Tools, you are required to add the official drivers for the databases to the local directory. The directory location to the drivers is `TIBCO_HOME\bw\palettes\smartmapper\7.1\tools\lib`.

Configuring Database Connection

Configure the database before running SmartMapper command-line tools, if you want to save the manual typing works for all parameters each time you connect to the database.

TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper provides a `db.properties` file to set up database configurations. Therefore, each time the SmartMapper tools are run, you are only required to enter the database alias which points to the database that you want to use.

Procedure

1. Locate the `db.properties` file in the `TIBCO_HOME\bw\palettes\smartmapper\7.1\tools` directory.
2. Open the `db.properties` file. The supported database types are MySQL, Oracle, and Microsoft SQL Server. For more details, see the readme file.
3. Set up database connection parameters according to the information listed in the following table.

Database Property	Description
*.type	The type of the database.
*.driver	The name of the driver class.
*.url	The URL to use to connect to the database.
*.username	The username to use when connecting to the database.
*.password	The password to use when connecting to the database.



The asterisk (*) represents the alias of the database. It can be any character you want to use and cannot be omitted, which is used to locate the database. In the given example, the database alias is `mysql`.

4. Save your configurations.

Using Bulk Extract Tool

With this tool, you can extract the relationship data or the entity data in bulk.

Prerequisites

Ensure that the **SmartMapper Tools** component is selected when installing the plug-in.

Procedure

1. On a command line, navigate to the directory in which the SmartMapper tools are located.
 - On Windows, the directory is `TIBCO_HOME\bw\palettes\smartmapper\7.1\tools`.
 - On Linux, the directory is `TIBCO_HOME/bw/palettes/smartmapper/7.1/tools`.
 - On Mac OS, the directory is `TIBCO_HOME/bw/palettes/smartmapper/7.1/tools`.

2. Enter `bulkextract` on the command line and press Enter to get help information.

3. Enter the following command to load the desired entity or relationship:

```
bulkextract -db <db config> -f <filename> -r <relname> | -e <entityname> [-c <criteria>]
```



On Linux and Mac OS, if you use Microsoft SQL Server to run the tool, make sure you enclose the database URL in quotation marks ("), unless you use the `db.properties` file to [configure the database connection](#).

See more information in [Parameters for SmartMapper Tools](#).



Entities (-e) and relationships (-r) cannot be extracted simultaneously. That is, each time you run this tool, only one type of data can be extracted, either entity or relationship.

Result

A log file named `bulk_extract.log` is generated in the `TIBCO_HOME\bw\palettes\smartmapper\7.1\tools\logs` directory after running the tool.

Using Bulk Load Tool

With this tool, you can load the relationship data or the entity data in bulk.

Prerequisites

Ensure that the **SmartMapper Tools** component is selected when installing the plug-in.



Ensure that the data to be loaded conform to rules described in [Data Format](#).

Procedure

1. On a command line, navigate to the directory in which the SmartMapper tools are located.
 - On Windows, the directory is `TIBCO_HOME\bw\palettes\smartmapper\7.1\tools`.
 - On Linux, the directory is `TIBCO_HOME/bw/palettes/smartmapper/7.1/tools`.

- On Mac OS, the directory is `TIBCO_HOME/bw/palettes/smartmapper/7.1/tools`.
2. Enter `bulkload` on the command line and press Enter to get help information.
 3. Enter the following command to export the desired entity or relationship:

```
bulkload -db <db config> -f <filename> -r | -e
```



On Linux and Mac OS, if you use Microsoft SQL Server to run the tool, make sure you enclose the database URL in quotation marks ("), unless you use the `db.properties` file to [configure the database connection](#).

See more information in [Parameters for SmartMapper Tools](#).



The entity and relationship data cannot be loaded simultaneously. That is, each time you run this tool, only one type of data can be loaded, either entity or relationship.

Result

A log file named `bulk_load.log` is generated in the `TIBCO_HOME\bw\palettes\smartmapper\7.1\tools\logs` directory after running the tool.

Data Format

When using Bulk Load tool, the data to load must conform to the specific rules.

- The data are in JSON format.
- One JSON object is one line.
- One file contains one type of data, either entity or relationship. The file cannot contain both the entity data and the relationship data.

Relationship Data Format

The following syntax shows the format of relationship data:

```
{ "name": "relationship_name", "participants": [ { "name": "participant_name", "fields": { "key1": "value1", "key2": "value2", "key3": "value3" } }, { "name": "participant_name", "fields": { "id": "value" } } ] }
```



If the participant is a TIBCO-Managed entity, the participant must be `{ "id": "value" }`. The plug-in generates the entity with the specified value. If no value is specified, the plug-in randomly assigns a value for the entity.

For example:

```
{ "name": "R1", "participants": [ { "name": "A", "fields": { "key1": "a1", "key2": "a2", "key3": "a3" } }, { "name": "B", "fields": { "id": "" } } ] }
```

Entity Data Format

The following syntax shows the format of entity data:

```
{ "name": "entity_name", "fields": { "key1": "value1", "key2": "value2", "key3": "value3" } }
```

For example:

```
{ "name": "A", "fields": { "key1": "a1", "key2": "a2", "key3": "a3" } }
```

Using Differencing Tool

With this tool, you can compare two SmartMapper ER Models.

Prerequisites

Ensure that the **SmartMapper Tools** component is selected when installing the plug-in.

Procedure

1. On a command line, navigate to the directory in which the SmartMapper tools are located.
 - On Windows, the directory is `TIBCO_HOME\bw\palettes\smartmapper\7.1\tools`.
 - On Linux, the directory is `TIBCO_HOME/bw/palettes/smartmapper/7.1/tools`.
 - On Mac OS, the directory is `TIBCO_HOME/bw/palettes/smartmapper/7.1/tools`.
2. Enter `diff` on the command line and press Enter to get help information.
3. Enter the following command to compare two SmartMapper ER Models:

```
diff -src [ear <EAR file directory!jar file directory!ER Model file path>] |
[file <ER Model file path>] | [db <type> <driver> <url> <username> <password>] |
[db < alias>] | [jar <jar file directory!ER Model file path>] -dest [ear <EAR
file directory!jar file directory!ER Model file path>] | [file <ER Model file
path>] | [db <type> <driver> <url> <username> <password>] | [db < alias>] | [jar
<jar file directory!ER Model file path>]
```



On Linux and Mac OS, if you use Microsoft SQL Server to run the tool, make sure you enclose the database URL in quotation marks ("), unless you use the `db.properties` file to [configure the database connection](#).



On Linux and Mac OS, add backslashes (\) before exclamation points (!) to separate *EAR file directory* and *JAR file directory* from *ER Model file* in this command.

For example:

```
diff -src file /opt/test.smartmapperermodel -dest ear /opt/test.ear\!
test.jar\!Resources/project/SmartMapperERModel.smartmapperermodel
```

See more information in [Parameters for SmartMapper Tools](#).

Result

A log file named `diff.log` is generated in the `TIBCO_HOME\bw\palettes\smartmapper\7.1\tools\logs` directory after running the tool.

Using Migration Tool

With this tool, you can migrate relationship data and entity data in an existing database to another empty database.

Prerequisites

Ensure that the **SmartMapper Tools** component is selected when installing the plug-in.



Ensure that the database to which the data is migrated is empty.

Procedure

1. On a command line, navigate to the directory in which the SmartMapper tools are located.
 - On Windows, the directory is `TIBCO_HOME\bw\palettes\smartmapper\7.1\tools`.
 - On Linux, the directory is `TIBCO_HOME/bw/palettes/smartmapper/7.1/tools`.
 - On Mac OS, the directory is `TIBCO_HOME/bw/palettes/smartmapper/7.1/tools`.
2. Enter `migrate` on the command line and press Enter to get help information.

3. Enter the following command to migrate the cross-referencing data from the source database to the destination database:

```
migrate -src <db config> -dest <db config>
```



On Linux and Mac OS, if you use Microsoft SQL Server to run the tool, make sure you enclose the database URL in quotation marks ("), unless you use the `db.properties` file to [configure the database connection](#).

See more information in [Parameters for SmartMapper Tools](#).

Result

A log file named `migrate.log` is generated under the `TIBCO_HOME\bw\palettes\smartmapper\7.1\tools\logs` directory after running the tool.

Using Database Upgrade Tool

With this tool, you can upgrade a database used for TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper 5.x. After upgrading, the database can be used in TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper 7.1.

Prerequisites

Ensure that the **SmartMapper Tools** component is selected when installing the plug-in.



Ensure that the 5.x database is synchronized and backed up.

Procedure

1. On a command line, navigate to the directory in which the SmartMapper tools are located.
 - On Windows, the directory is `TIBCO_HOME\bw\palettes\smartmapper\7.1\tools`.
 - On Linux, the directory is `TIBCO_HOME/bw/palettes/smartmapper/7.1/tools`.
 - On Mac OS, the directory is `TIBCO_HOME/bw/palettes/smartmapper/7.1/tools`.
2. Enter `upgrade` on the command line and press Enter to get help information.
3. Enter the following command to migrate the cross-referencing data from the source database to the destination database:

```
upgrade -db <db config> -ermodel <ermodel file>
```



On Linux and Mac OS, if you use Microsoft SQL Server to run the tool, make sure you enclose the database URL in quotation marks ("), unless you use the `db.properties` file to [configure the database connection](#).

See more information in [Parameters for SmartMapper Tools](#).




Result




A log file named `upgrade.log` is generated under the `TIBCO_HOME\bw\palettes\smartmapper\7.1\tools\logs` directory after running the tool.





Parameters for SmartMapper Tools

To use SmartMapper tools, you have to configure the commands with proper parameters.

The following table explains the parameters used in the tools:

Parameter	Comment
-db	<p>Specify the database connection.</p> <p>The value of this parameter is either the database alias configured in the <code>db.properties</code> file, or a group of parameters used to connect to the database. If you do not use the database alias, the following database related parameters must be specified: type, driver, URL, username, and password.</p> <p>For example:</p> <pre>-db mysql com.mysql.jdbc.Driver jdbc:mysql://localhost:3306/test root admin</pre>
-f	Specify the name and the path of the file where the data are stored.
-e	Specify the name of the entity.
-r	Specify the name of the relationship.
-c	<p>Specify a criterion to filter the entity instance or the relationship instance that you want to export. This parameter is optional.</p> <p>When multiple filter criteria are provided, each of them is AND'ed if operator and value are specified using the where clause. Conditions are OR'ed together if operator and value are specified using the orWhere clause.</p> <p>A filter criterion consists of three parts, which are separated by commas (,). The first part is the filter condition, where or orWhere; the second part is the entity's field name, or the participant name and the corresponding field name separated by a dot (.); the third part is the value of the field.</p> <p>Use semicolon (;) to separate multiple filter criteria.</p> <div>  <p>When adding two filters and each filter is a different participant, the filters are only combined as OR; and if one filter is a participant and the other filter is a createTimestamp, the two filters are also only combined as OR.</p> </div> <div>  <p>When adding filters for timestamps, the filter names must only be createTimestamp or updateTimestamp.</p> <p>For example:</p> <pre>-c "where, createTimestamp,>=2013-09-10; where, createTimestamp,<=2014-05-15"</pre> <p>For example:</p> <p>Filter entity data:</p> <pre>-c "where, key1, like '%1%'; where, key1, != 'a21' "</pre> <p>Filter relationship data:</p> <pre>-c "where, A. key1, = 'a11' ; orWhere, C. key1, = 'c1' "</pre> <div>  <p>This parameter is used only in Bulk Extract tool.</p> </div> </div>

Parameter	Comment
-ermodel	<p>Specify the SmartMapper ER Model.</p> <p>The value of this parameter is the path of the ER Model file, which is detailed to the SmartMapper ER Model name.</p> <p>For example:</p> <pre>upgrade -db mysql -ermodel c:\SmartMapperERModel.smartmapperermodel</pre> <p> This parameter is used only in Database Upgrade Tool.</p>
-src	<p>Specify the source SmartMapper ER Model.</p> <p>The value of this parameter is either the database alias configured in the <code>db.properties</code> file, or a group of parameters used to connect to the database. If you do not use the database alias, the following database related parameters must be specified: type, driver, URL, username, and password.</p> <p> This parameter is used in Migration Tool and Differencing Tool.</p>
-dest	<p>Specify the destination SmartMapper ER Model.</p> <p>The value of this parameter is either the database alias configured in the <code>db.properties</code> file, or a group of parameters used to connect to the database. If you do not use the database alias, the following database related parameters must be specified: type, driver, URL, username, and password.</p> <p> This parameter is used only in Migration Tool and Differencing Tool.</p>
When using the Differencing Tool , the following parameters are combined with -src and -dest parameters.	
db	<p>Specify a database in which the SmartMapper ER Model is to be compared.</p> <p>The value of this parameter is either the database alias configured in the <code>db.properties</code> file, or a group of parameters used to connect to the database. If you do not use the database alias, the following database related parameters must be specified: type, driver, URL, username, and password.</p> <p>For example:</p> <pre>diff -src mysql com.mysql.jdbc.Driver jdbc:mysql://localhost:3306/ test root admin</pre>

Parameter	Comment
ear	<p>Specify an Enterprise Archive (EAR) file in which the SmartMapper ER Model is to be compared.</p> <p>The value of this parameter consists of two parts that are separated by an exclamation mark (!). The first part is the path of the EAR file; the second part is the folder where the specified SmartMapper ER Model is located, which is detailed to the SmartMapper ER Model name.</p> <div>  <p>If the file name or the file path contains spaces, enclose the value in the quotation marks ("").</p> </div> <div>  <p>On Linux and Mac OS, add backslashes (\) before exclamation points (!) to separate <i>EAR file directory</i> from <i>ER Model file</i> in this command.</p> </div> <p>For example:</p> <pre>diff -src file c:/test.smartmapperermodel -dest ear c:/test.ear! test.jar!Resources/project/SmartMapperERModel.smartmapperermodel</pre>
jar	<p>Specify a Java ARchive (JAR) file in which the SmartMapper ER Model to be compared.</p> <p>The value of this parameter consists of two parts that are separated by an exclamation mark (!). The first part is the path of the JAR file; the second part is the folder where the specified SmartMapper ER Model is located, which is detailed to the SmartMapper ER Model name.</p> <div>  <p>On Linux and Mac OS, add backslashes (\) before exclamation points (!) to separate <i>JAR file directory</i> from <i>ER Model file</i> in this command.</p> </div> <p>For example:</p> <pre>diff -src jar c:/test.jar!Resources/project/ SmartMapperERModel.smartmapperermodel -dest file c:/ SmartMapperERModel.smartmapperermodel</pre>
file	<p>Specify a SmartMapper ER Model file to be compared.</p> <p>The value of this parameter is the path of the SmartMapper ER Model file, which is detailed to the SmartMapper ER Model name.</p> <div>  <p>If the file name or the file path contains spaces, enclose the value in the quotation marks ("").</p> </div> <p>For example:</p> <pre>diff -src file c:/test.smartmapperermodel -dest ear c:/test.ear! test.jar!Resources/project/SmartMapperERModel.smartmapperermodel</pre>

Working with Sample Projects

Working through the sample projects helps you understand how TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper works.

The plug-in packages three sample projects within the installer. The following sample projects are located in the `TIBCO_HOME\bw\palettes\smartmapper\7.1\examples` directory:

- The [Getting_Started Project](#) shows how to seed data to the SmartMapper ER Model and look up relationship data.
- The [Wizard Project](#) shows how to use the SmartMapper Wizard activity to perform a batch of SmartMapper operations.
- The [BulkExtractFilter Project](#) shows how to extract and load relationship data in a bulk.

Importing Sample Projects

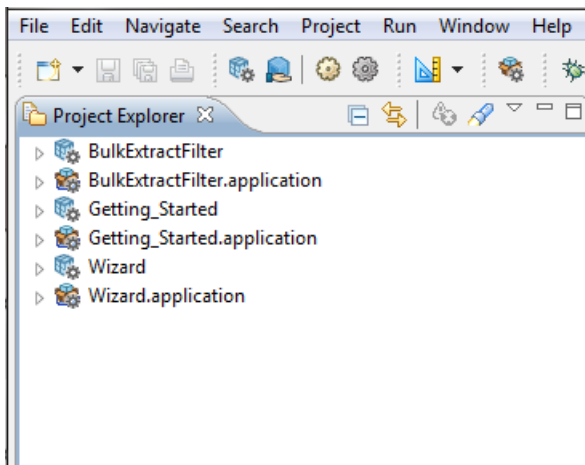
Before running the sample project, you must import the project to TIBCO Business Studio.

Procedure

1. Start TIBCO Business Studio 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 the **Existing Studio Projects into Workspace** item. Click **Next**.
4. Click **Browse** next to the **Select root directory** field to locate the sample. Click **Finish**.
The sample project is located in the `TIBCO_HOME/bw/palettes/smartmapper/7.1/examples` directory.

Result

The sample projects are imported to TIBCO Business Studio.



Working with Getting_Started Project

This sample project shows how to seed data to the SmartMapper ER Model and look up relationship data.

Business Scenario

An inbound message (using Siebel application) needs to be looked up and seeded to an outbound message to another data model (using OAG-based application).

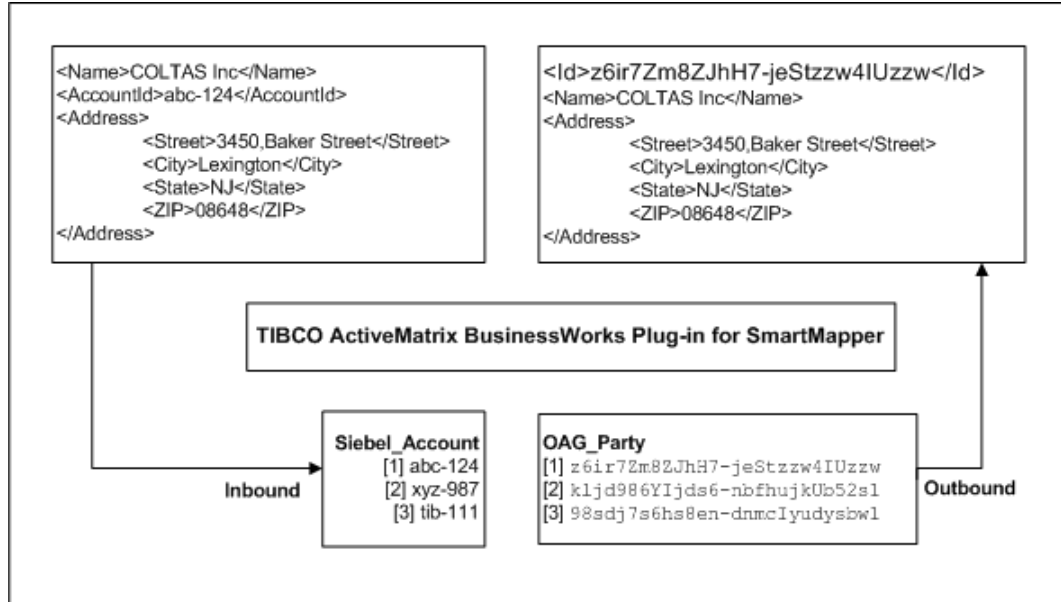
The inbound message from a Siebel application references a Siebel customer through a Siebel_Account_ID field. Some of the data in this inbound message must be used to generate an outbound message to an OAG-based common data model. This OAG-based common data model uses an OAG_Party_ID field to refer to customers.

The inbound Siebel_Account_ID needs to be cross-referenced with the corresponding OAG_Party_ID on the OAG-based common data model at run time. Also, if such a cross-reference does not exist, a new OAG_Party_ID needs to be created and cross-referenced with the inbound Siebel_Account_ID for possible future use.

TIBCO ActiveMatrix BusinessWorks Plug-in for SmartMapper performs the necessary cross-referencing required by the business scenario. The Siebel_Account_ID field is cross-referenced with the OAG_Party_ID field. When an inbound Siebel account is received by the SmartMapper cross-referencing engine at run time, it is cross-referenced with the corresponding OAG_Party_ID, which is returned as output.

For entity values that do not exist in an underlying data store, SmartMapper dynamically generates an OAG_Party_ID, and then creates a cross-reference in the model at run time.

The following figure shows the whole solution:

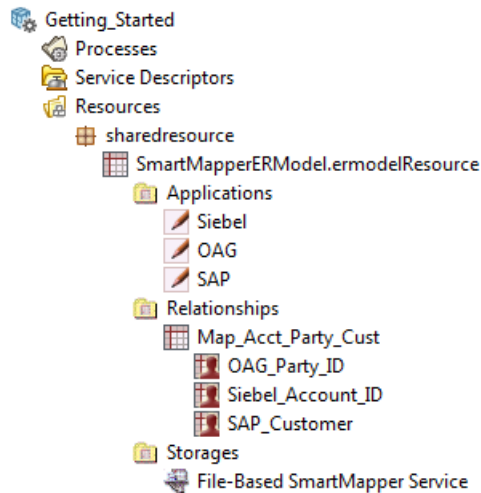


SmartMapper ER Model

The SmartMapper ER Model resource represents a logical data model that contains a set of applications, relationships and storage. A SmartMapper ER Model needs to be created before creating a SmartMapper process.

Three applications and one relationship are defined in the SmartMapper ER Model as to demonstrate this sample project, a file-based SmartMapper service is used to store the relationship data. Each application has an associated entity and each entity has a defined data schema.

The following figure shows the complete SmartMapper ER Model:



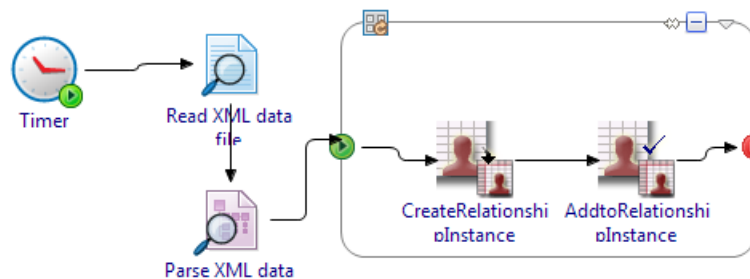
Configuration for Getting_Started Project

This project shows how to seed data to the SmartMapper ER Model and to look-up relationship data.

This project contains two processes: Seed All Data into Model and Lookup Data.

Seed All Data into Model process

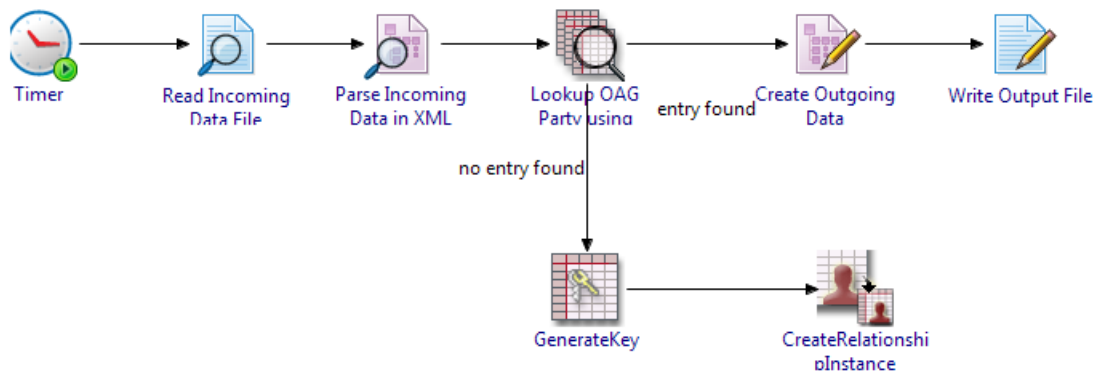
This process demonstrates how to seed relationship data to the created SmartMapper ER Model. The following figure shows the composing activities:



Activity	Description
Read XML data file	Reads the pre-defined file where contains the relationship data. The Seed_Data.xml file is the input.
Parse XML data	Parses the XML file. The output of the Read XML data file activity is the input.
CreateRelationship Instance	Creates a relationship instance for the Map_Acct_Party_Cust relationship defined in the SmartMapper ER Model. The Siebel_Account_ID and the OAG_Party_ID are the input.
AddtoRelationship Instance	Adds participants to the relationship instance added by using the Create Relationship Instance activity.

Lookup Data process

This process shows how to look up Siebel_Account_ID in the cross-referenced tables.



Activity	Description
Read Incoming Data File	Reads Siebel_Account.xml based on input field fileName and generates xmlString in output field textContent.
Parse Incoming Data in XML	Parses the xmlString field based on the XMLschema. Output Editor tab: the Siebel_Account.xsd schema in the Schema field is used to validate the contents of the Siebel_Account.xml file that the Read Incoming Data File activity produces.
Look-up OAG Party using Siebel_Account Activity	Gets id in the Input tab is configured to point to the AccountId property of the Siebel_Account. The AccountId is the output of the Parse Incoming Data In XML activity. The output for this activity is the OAG_Party_ID key that is returned by the mapping service.
Generate Key	If the lookup fails, a new key is generated. When the user runs the process for a second time, the process can succeed because the key has been generated.
Create Relationship Instance	If the lookup fails, a new relationship instance is created.
Create Outgoing Data	If the lookup returns a value, a new XML file is created for the outgoing mapping data.
Write Output File	If the lookup returns a value, the mapping data is written into the XML file that is generated in Create Outgoing Data activity.


Running Getting_Started Application

Run the Getting_Started Application in TIBCO Business Studio to see the cross-referenced data .

Prerequisites

Before running the project, ensure that you have imported the sample project to TIBCO Business Studio, as described in [Importing Sample Projects](#).

Procedure

1. Expand the Getting_Started project in the Project Explorer view.
2. Click **Module Descriptors > Module Properties**.
3. Set the value of property **SmartMapper.demo.directory** to *TIBCO_HOME\bw\palettes\smartmapper\7.1\examples\Getting_Started\Getting_Started*.
4. Set the value of property **file_based_SmartMapper** to *TIBCO_HOME\bw\palettes\smartmapper\7.1\examples\Getting_Started\Getting_Started\SmartMapper-SERVICE.xml*.
5. Click **Run > Run Configurations**.
6. Expand **BusinessWorks Application** in the Run Configurations window and click **BWApplication**.
7. Click **Deselect All** in the **Applications** tab and select the check box next to the Getting_Started.application project.
8. Click **Run**.
9. Click the **Terminate**  icon to terminate the process.

Working with Wizard Project

This sample project shows how to use the SmartMapper Wizard activity to perform a batch of SmartMapper operations.

Business Scenario

When there are many things to cross-reference, because there are repeated elements, the SmartMapper Wizard is preferred.

An order is mapped to a common data model so that it might be published out to the application. Some subscribers might be interested in doing billing, shipping, provisioning, or planning. All subscribers must know that a new sales order was booked.

Because there are many things to cross-reference on a sales order, and because there are repeated elements like order lines, the SmartMapper Wizard is used.

SmartMapper Solution

The SmartMapper Wizard maps several fields to produce output that is used to generate an OAG_SalesOrder.

The following steps show an overview of how SmartMapper works:

1. The inbound message is taken from a Siebel application.

```
<OrderNumber>100</OrderNumber>
<AccountId>abc-123</AccountId>
<OrderDate>2003-01-31</OrderDate>
<LineItems>
  <LineNumber>10</LineNumber>
  <ProductId>101111</ProductId>
  <Quantity>3</Quantity>
</LineItems>
<LineItems>
  <LineNumber>20</LineNumber>
  <ProductId>101048</ProductId>
  <Quantity>1</Quantity>
</LineItems>
```

2. The inbound order number is cross-referenced with the generated order number.

OrderNumber [1] 100	ID [1] cross-referenced with remote ID: 1YrEz4qfZMI4v-Hjdxzzw4r-zzw
AccountId [1] abc-123	PartyId [1] z6ir7Zm8ZJhH7-jeStzzw4IUzzw cross-referenced with remote PartyId: Wc5jtvOUZMI4xUHjdyzzw4r-zzw

3. An outbound message is created to be understood by other services on the integration backbone.

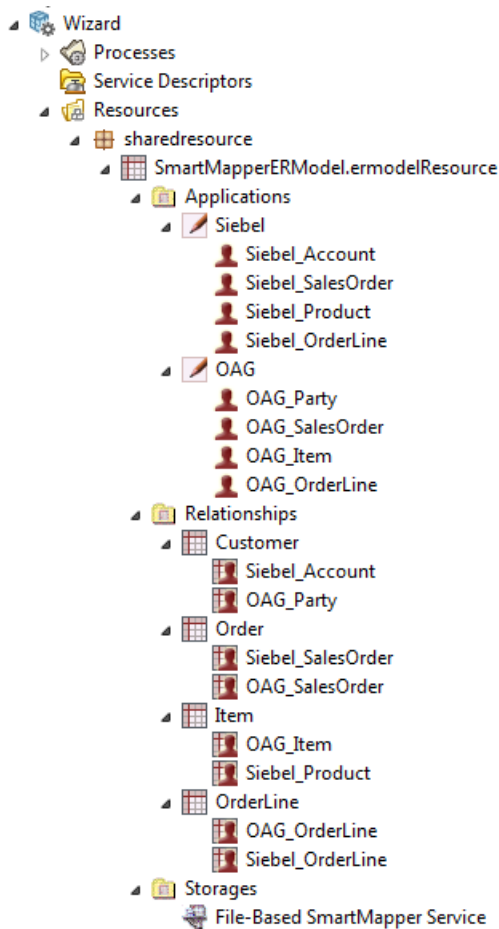
```
<Id>1YrEz4qfZMI4v-Hjdxzzw4r-zzw</Id>
<PartyId>Wc5jtvOUZMI4xUHjdyzzw4r-zzw</PartyId>
<DocumentDateTime>2003-01-31</DocumentDateTime>
<Line>
  <LineNumber>QvmSp5B3ZMI4xUHjdzzw4r-zzw</LineNumber>
  <ItemId>cd6Nfd5IZMI4xUHje-zzw4r-zzw</ItemId>
  <OrderQuantity>3</OrderQuantity>
</Line>
```

SmartMapper ER Model

The SmartMapper ER Model resource represents a logical data model that contains a set of applications, relationships and storage. A SmartMapper ER Model needs to be created before creating a SmartMapper process.

The SmartMapper ER Model used in this Wizard project contains two applications and four relationships. A file-based SmartMapper service is used to store the relationship data. Four entities are added in each application and each entity has its own data schema.

The following figure shows the complete SmartMapper ER Model:

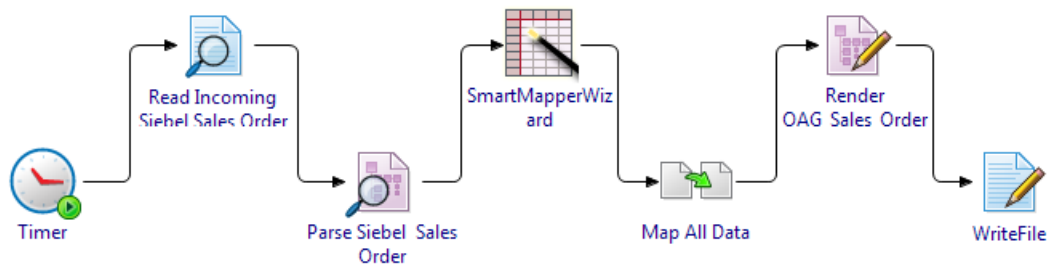


Configuration for Wizard Project

This project shows how to use SmartMapper to link the inbound message and the remote system. One process is contained in this project: Create Sales Order.

Create Sales Order

This process demonstrates how to create a corresponding order on a remote system at run time when a new inbound sales order arrives. The following figure shows the composing activities:



Activity	Description
Reading Incoming Siebel Sales Order	Reads the pre-defined file where contains the relationship data. The Siebel_SalesOrder.xml file is the input.

Activity	Description
Parse Siebel Sales Order	Parses the schema data into XML data. The output of the Reading Incoming Siebel Sales Order activity is the input.
SmartMapperWizard	Looks up on the parsed XML data and generates keys and field in the OAG application. See SmartMapper Wizard see more details about this activity.
Map All Data	Creates a mapping file for the OAG sales order based on what SmartMapperWizard has generated.
Render OAG Sales Order	Renders the XML schema as a XMLString in output.
WriteFile	Writes the XMLString into OAG_Sales_Order_XML.


Running Wizard Application

Run the Wizard Application in TIBCO Business Studio to see the mapped data.

Prerequisites

Before running the project, ensure that you have imported the sample project to TIBCO Business Studio, as described in [Importing Sample Projects](#).

Procedure

1. Expand the Wizard project in the Project Explorer view.
2. Click **Module Descriptors > Module Properties**.
3. Set the default value of property **SmartMapper.demo.directory** to `TIBCO_HOME\bw\palettes\smartmapper\7.1\examples\Wizard\Wizard`.
4. Set the default value of property **file_based_SmartMapper** to `TIBCO_HOME\bw\palettes\smartmapper\7.1\examples\Wizard\Wizard\SmartMapper-Service.xml`.
5. Click **Run > Run Configurations**.
6. Expand **BusinessWorks Application** in the Run Configurations window and click **BWApplication**.
7. Click **Deselect All** in the **Applications** tab and select the check box next to the Wizard.application project.
8. Click **Run**.
9. Click the **Terminate**  icon to terminate the process.

Working with BulkExtractFilter Project

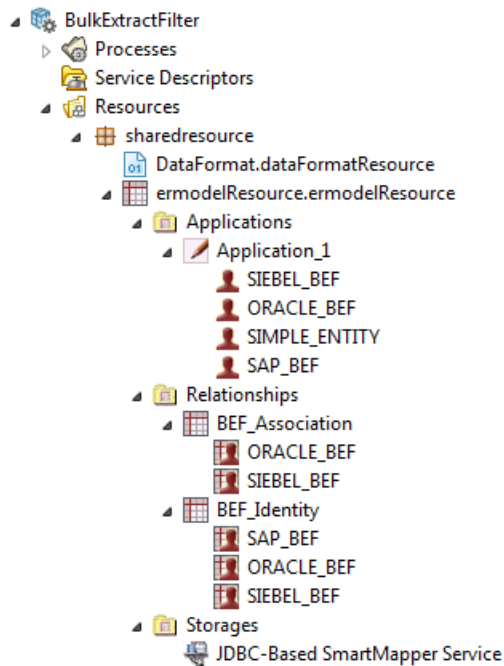
This sample project shows how to extract and load relationship data in a bulk. Two processes are pre-configured in this project, LookupData process and SeedAllDataintomodel process.

SmartMapper ER Model

The SmartMapper ER Model resource represents a logical data model that contains a set of applications, relationships and storage. A SmartMapper ER Model needs to be created before creating a SmartMapper process.

The SmartMapper ER Model used in this project contains one application and two relationships. A JDBC-Based SmartMapper service is used to store the relationship data. Four entities are added in the application and each entity has its own data schema.

The following figure shows the complete SmartMapper ER Model:

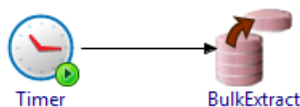


Configuration for BulkExtractFilter Project

This project shows how to use the Bulk Extract activity to look-up relationship data based on different criteria. The project includes three process folders: association, entity and identity.

Bulk Extract process

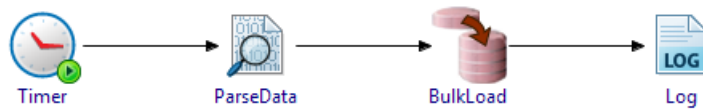
Bulk Extract process extracts all the relationship data or the entity data. This process is included in all three process folders. The following figure shows the composing activities:



Activity	Description
BulkExtract	Extracts data based on criteria, see Bulk Extract for more information.

Bulk Load process

Bulk Load process loads the specified sample data to the SmartMapper database. This process is included in all three process folders. The following figure shows the composing activities:



Activity	Description
ParseData	Parse data file based on the XML schema file.
BulkLoad	Loads the specified data to the SmartMapper database, see Bulk Load for more information.
Log	Writes the output messages of the BulkLoad activity and print the messages in the console when running the process.

Running BulkExtractFilter Application


Run the BulkExtractFilter Application in TIBCO Business Studio to see the extracted data.

Prerequisites

Before running the project, ensure that you have imported the sample project to TIBCO Business Studio, as described in [Importing Sample Projects](#).

Procedure

1. Expand the BulkExtractFilter project in the Project Explorer view.
2. Click **Module Descriptors > Module Properties**.
3. Set the default value of property **SM_HOME** to *TIBCO_HOME\bw\palettes\smartmapper\7.1*.
4. Set the default values of **JDBC_Driver**, **JDBC_PASSWORD**, **JDBC_USER** and **JDBC_URL** according to the actual JDBC-Based SmartMapper Service.
5. Click **Resources > sharedresource > ermodelResource.smartmapperermodel > Storages**.
6. In the **JDBC Settings** tab, click **Sync ER Model w/DB**.
7. In the **Sync To DB** tab, select the **Audit** check box to enable auditing.
8. Use one of the following ways to perform the synchronization:
 - Click **Generate SQL** or **Generate Complete Script** to copy a SQL script and run the script in your database.
 - Click **Perform Sync** to start the synchronization.

9. Close the dialog when the synchronization completes.
10. Click **Run > Run Configurations**.
11. Expand **BusinessWorks Application** in the Run Configurations window and click **BWApplication**.
12. Click **Deselect All** in the **Applications** tab and select the check box next to the BulkExtractFilter.application project.
13. Click **Run**.
14. Click the **Terminate**  icon to terminate the process.

Managing Logs

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

By default, error logs are displayed in the Console view when you run a process. 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
Trace	Includes all information regarding the running process.
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.
Error	Indicates that an unrecoverable error has occurred. Depending on the error severity, the plug-in might continue with the next operation or might stop altogether.
Debug	Indicates a developer-defined tracing message.

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 log level configured for TIBCO ActiveMatrix BusinessWorks. The default log level value 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.smartmapper.runtime">
  <level value="DEBUG"/>
</logger>
```

The value of the `level` element can be ERROR, INFO, DEBUG, or TRACE.



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. The node varies according to the activity and the palette that it belongs to.

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

For example, you can add the following node to set the log level of the Bulk Extract activity to DEBUG:

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



The activities that are not configured with specific log levels use the log level configured for 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/space_name/node_name` directory to find the `logback.xml` file.

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

```
<appender name="FILE" class="ch.qos.logback.core.FileAppender">
  <file>c:/bw6-adsmartmapper.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 log.

3. Add the following node to 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.

Error Codes

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

Error Code and Error Message	Role	Category	Description	Solution
TIBCO-BW-PALETTE-SMARTMAPPER-500002 IOException occurred while retrieving XML Output for activity [{0}].	error Role	BW-Plug-in	Fails to retrieve XML Output for activity.	No action.
TIBCO-BW-PALETTE-SMARTMAPPER-500004 Participant is not specified.	error Role	BW-Plug-in	No participant is selected.	Select a participant to use.
TIBCO-BW-PALETTE-SMARTMAPPER-500005 Entity is not specified.	error Role	BW-Plug-in	No entity is selected.	Select an entity to use.
TIBCO-BW-PALETTE-SMARTMAPPER-500006 Relationship is not specified.	error Role	BW-Plug-in	No relationship is selected.	Select a relationship to use.
TIBCO-BW-PALETTE-SMARTMAPPER-500007 Input participant is not specified.	error Role	BW-Plug-in	No participant is specified in the Input Participant field.	Specify a participant to use in the Input Participant field.
TIBCO-BW-PALETTE-SMARTMAPPER-500008 Output participant is not specified.	error Role	BW-Plug-in	No participant is specified in the Output Participant field.	Specify a participant to use in the Output Participant field.
TIBCO-BW-PALETTE-SMARTMAPPER-500009 Could not create the SmartMapper service for: [{0}].	error Role	BW-Plug-in	Fails to create the SmartMapper storage service.	Check the configurations of the storage service.

Error Code and Error Message	Role	Category	Description	Solution
TIBCO-BW-PALETTE-SMARTMAPPER-500011 Could not initialize Relationship, [{0}]. Perhaps the SmartMapper ER Model has not been synchronized with the data store.	error Role	BW-Plug-in	Fails to initialize the specified relationship.	Check whether the SmartMapper ER Model has been synchronized.
TIBCO-BW-PALETTE-SMARTMAPPER-500013 Wizard Operation Exception: [{0}].	error Role	BW-Plug-in	Fails to run the process when using SmartMapper Wizard activity.	Check the configurations of the SmartMapper Wizard activity.
TIBCO-BW-PALETTE-SMARTMAPPER-500015 SmartMapper Exception: [{0}].	error Role	BW-Plug-in	Fails to run the process.	Check the configurations.
TIBCO-BW-PALETTE-SMARTMAPPER-500017 Invalid input data.	error Role	BW-Plug-in	The input data is invalid.	Check your input configurations.
TIBCO-BW-PALETTE-SMARTMAPPER-500018 Unsupported operation: Entity [{0}] has non-keys. Non key attributes are not supported for File based service.	error Role	BW-Plug-in	The entity has non-key field specified. The file-based service does not support non-key attributes.	Change a SmartMapper storage service, or change the non-key schema to key.
TIBCO-BW-PALETTE-SMARTMAPPER-500019 No active service found.	error Role	BW-Plug-in	No SmartMapper storage service is active.	Activate a SmartMapper storage service to store the cross-referencing data.
TIBCO-BW-PALETTE-SMARTMAPPER-500020 Participant, [{0}] not found, please make sure participant name matches with the one defined in the SmartMapper ER Model.	error Role	BW-Plug-in	Fails to find the specified participant.	Check the name of the participant. Ensure that the name is the same with the one defined in the SmartMapper ER Model.

Error Code and Error Message	Role	Category	Description	Solution
TIBCO-BW-PALETTE-SMARTMAPPER-500021 Relationship, [{0}] not found, please make sure relationship name matches with the one defined in the SmartMapper ER Model.	error Role	BW-Plug-in	Fails to find the specified relationship.	Check the name of the relationship. Ensure that the name is the same with the one defined in the SmartMapper ER Model.
TIBCO-BW-PALETTE-SMARTMAPPER-500022 Invalid subsetsize value : [{0}], value must be greater than 0.	error Role	BW-Plug-in	The value for the subset is invalid.	Enter an integer greater than 0.
TIBCO-BW-PALETTE-SMARTMAPPER-500024 [{0}] format cannot be used with [{1}] relationship type.	error Role	BW-Plug-in	The data format does not match with the relationship type.	Check your configurations. If the relationship has one participant, then identity relationship must be selected; if the relationship has only two participants, then association relationship must be selected.
TIBCO-BW-PALETTE-SMARTMAPPER-500025 Participant, [{0}] not found for relationship [{1}] please make sure participant name matches with the one defined in the SmartMapper ER Model.	error Role	BW-Plug-in	Fails to find the relationship with the specified participant.	Check the name of the participant. Ensure that the name is the same as the one defined in the SmartMapper ER Model.
TIBCO-BW-PALETTE-SMARTMAPPER-500026 ER Model cannot be empty.	error Role	BW-Plug-in	The SmartMapper ER Model fails to be empty.	Check your configuration.
TIBCO-BW-PALETTE-SMARTMAPPER-500027 Unsupported operation: File based service is not supported for this activity.	error Role	BW-Plug-in	The file-based service does not support the current operation.	Change a SmartMapper storage service.

Error Code and Error Message	Role	Category	Description	Solution
TIBCO-BW-PALETTE-SMARTMAPPER-500028 Schema is not defined.	error Role	BW-Plug-in	No schema is defined for the specified entity.	Define a schema for the entity.
TIBCO-BW-PALETTE-SMARTMAPPER-500030 Attribute index value, [{0}], is out of valid indexes range ([{1}]) for this participant.	error Role	BW-Plug-in	The entered value is invalid.	Enter a valid value.
TIBCO-BW-PALETTE-SMARTMAPPER-500031 Relationship instance creation requires at least two participants, only [{0}] is provided in the input.	error Role	BW-Plug-in	Only one participant is specified when creating the relationship instance.	Check your input configuration.
TIBCO-BW-PALETTE-SMARTMAPPER-500032 At least one participant, provided in the input, needs to be of cardinality one.	error Role	BW-Plug-in	No cardinality participant is specified.	Specify a cardinality participant.
TIBCO-BW-PALETTE-SMARTMAPPER-500033 Following required fields(name/index, value) are missing from input: [{0}].	error Role	BW-Plug-in	No value is specified for the required fields in the Input tab.	Check your input configurations.
TIBCO-BW-PALETTE-SMARTMAPPER-500038 One and only one participant can be flagged as lookup participant, currently [{0}] participants are flagged, please check the lookupParticipant flag value in input data.	error Role	BW-Plug-in	Only one participant is needed for lookup participant field.	Check your configurations.

Error Code and Error Message	Role	Category	Description	Solution
TIBCO-BW-PALETTE-SMARTMAPPER-500039 Attempt to update one cardinality participant [{0}], with many cardinality input; please check the value of unlinkParticipant flag in the input.	error Role	BW-Plug-in	The specified value is invalid in the Input field.	Check your configurations.
TIBCO-BW-PALETTE-SMARTMAPPER-500041 The schema [{0}] is not declared in Entity definition.	error Role	BW-Plug-in	The input value is invalid.	Enter a valid value.
TIBCO-BW-PALETTE-SMARTMAPPER-500042 No value is provided for participant [{0}].	error Role	BW-Plug-in	No value is provided for the participant.	Enter a value for the participant.
TIBCO-BW-PALETTE-SMARTMAPPER-500043 At least one field should be provided for participant [{0}].	error Role	BW-Plug-in	No field is provided for the participant.	Specify a field for the participant.
TIBCO-BW-PALETTE-SMARTMAPPER-500044 Incorrect use of flags in input data: [{0}].	error Role	BW-Plug-in	The input value is invalid.	Enter a valid value.
TIBCO-BW-PALETTE-SMARTMAPPER-500045 At least one field should be provided for entity [{0}].	error Role	BW-Plug-in	No field is provided for the entity.	Specify a field for the participant.
TIBCO-BW-PALETTE-SMARTMAPPER-500046 Could not initialize Entity, [{0}]. Perhaps the SmartMapper ER Model has not been synchronized with the data store.	error Role	BW-Plug-in	Fails to initialize the specified entity.	Check whether the SmartMapper ER Model has been synchronized.

Error Code and Error Message	Role	Category	Description	Solution
TIBCO-BW-PALETTE-SMARTMAPPER-500049 Entity [{0}] not found, please make sure entity name matches with the one defined in the ER Model.	error Role	BW-Plug-in	Fails to match the entity name with the entity name defined in the ER Model.	Check the entity name. Ensure the entity name matches with the one defined in the ER Model.
TIBCO-BW-PALETTE-SMARTMAPPER-500050 Field [{0}] not found in entity [{0}]	error Role	BW-Plug-in	Fails to find the specified field in the entity.	Check the entity name. Ensure the entity name matches with the one defined in the ER Model.
TIBCO-BW-PALETTE-SMARTMAPPER-500051 Invalid input schema, the input schema does not match the resource provided by Module Property.	error Role	BW-Plug-in	Fails to match the input schema with the resource provided by Module Property.	Provide a valid resource that matches the input schema.
TIBCO-BW-PALETTE-SMARTMAPPER-501999 Unexpected error occurs [{0}].	error Role	BW-Plug-in	Fails to initialize the process.	Check your configurations.