



Mapping Data Objects

Overview

Map data objects in the **Source** with objects in the **Destination** so that objects can easily be copied between the source and destination. Mapping data is applied to an object's referenced objects and not to the actual object itself. Mappings are also used to automate the copying process as discrepancies between source and destination object references are automatically mapped given the mapping rules. For convenience, mapping rules can be saved in a map file and restored later.

The data objects of the source do not always exist in the destination. These differences can be resolved by matching a data object in the source that does not exist in the destination to a different but comparable data object in the destination.

The mapping process is done in the **Scheduler Object Mapping** dialog. In this dialog, a list of the various types of data objects is divided into two columns. One column displays the data objects of the source and the other column displays the data objects of the destination master. By comparing the source and destination side-by-side, you can easily spot any differences that need to be mapped.

The mapping process involves going down the column of source data objects and ensuring that a match for the comparable data object that is missing in the destination is mapped. Once the mapping is completed, the copying process that transfers the jobs or other data types from one source to another can start. Mapping can also be done dynamically for Job transports during the actual copying procedure so that a complete map is built over time.

Scheduler Object Mapping Dialog

To display the **Scheduler Object Mapping** dialog, click the **Mappings** button on the Transporter toolbar or select the **Mappings** option of the Transport main menu.

The **Scheduler Object Mapping** dialog displays the following features:

- **Object Map Filename** – The name of the object map file. By default, the file is given the name of the source and given a *.map* extension.
- **Connections** – This tab displays a list of defined agent connections that exist in the source and provides for a mapping of a corresponding destination agent.
- **Agent Lists** – This tab displays a list of defined agent lists that exist in the source and provides for a mapping of a corresponding destination agent list.
- **Calendars** – This tab displays a list of defined calendars that exist in the source and provides for a mapping of a corresponding destination calendar.

**Note**

If a selected calendar depends upon a fiscal calendar, the fiscal calendar is also copied with the selected calendar.

- **Classes** – This tab displays a list of defined job classes that exist in the source and provides for a mapping of a corresponding destination job class.
- **Events** – This tab displays a list of defined events that exist in the source and provides for a mapping of a corresponding destination event.
- **Groups** – This tab displays a list of defined job groups that exist in the source and provides for a mapping of a corresponding destination group.
- **Owners** – This tab displays a list of defined users and workgroup owners that exist in the source and provides for a mapping of a corresponding destination owner.
- **Users** – This tab displays a list of defined users that exist in the source and provides for a mapping of a corresponding destination user.
- **Variables** – This tab displays a list of defined variables that exist in the source and provides for a mapping of a corresponding destination variable.
- **Virtual Resources** – This tab displays a list of defined virtual resources that exist in the source and provides for a mapping of a corresponding destination virtual resource.
- **Custom Resources** – This tab displays a list of defined custom resources that exist in the source and provides for a mapping of a corresponding destination custom resource.
- **Timezones** – This tab displays a list of defined timezones that exist in the source and provides for a mapping of a corresponding destination timezone.

How To Map Data Objects

Each list that is displayed by clicking one of the buttons listed above is divided into two columns that display the selected data object type from the source and destination. If an object type that exists in the source also exists in the destination master, it is displayed in the destination column (unless previously mapped to a different destination object). If there is no matching object type on the destination side (and that object has not been mapped), that space is blank.

On the right side of each **Destination** text field is a combo box that displays a list of available values. Select an object from the list to associate with the source data object. Ideally, you would match up each value displayed on the source side with a value on the destination side.

However, the jobs or other data that are being copied may not use all of the values that exist in the source. You may wish to map only the data object values that you think are needed and leave the rest unmapped.

**Note**

Mappings apply an object's referenced objects and not the actual object itself.

The idea behind mapping the data objects is to provide a means to automate copying of data between the source and destination.

With a map file, it is easy to schedule a batch job in TES to copy a job or job group between the source and destination. Refer to [“Running Transporter in Batch Mode”](#) for information on using Transporter in batch mode.

Default Mapping Option

You can configure a default mapping value for each type of data object. Each list of potential values for a data object type in the source starts with a default. This default has no value until it is assigned a destination value.

Specify a value for default by selecting a value from the list on the Destination side. This default value will be used during object copying whenever a corresponding data object is not mapped in the destination.

Not only can using a default value cut down on possible mapping errors, it can also save you considerable time. For example, if you are mapping connection objects and a large number of connections exist in the Source, but only one connection is used in the Destination, make the one connection the default. This avoids having to map each individual object. However, if the default value is incompatible with the type of data object being mapped, an error will still result. For example, if a connection object requires a Unix value and the default value is a Windows value, an error will result. In another example, an error would also result if a text string variable is required but the default value is a date calculation variable.

Remove Mapping Option

One of the options in the list on the destination side is **Remove**. If there is no corresponding destination data object to match an existing data object in the Source, you can just delete any reference to the data object when copying objects between Source and Destination. There is no **Remove** option on the source side. It only exists on the destination side.

The **Remove** option allows a source object to be copied to the destination. For example, jobs must have an owner and an agent so you cannot remove the owner or agent data objects.

**Note**

With the **Remove** option, data is transported to the destination, while “removing” object references that had been mapped to Remove. Note that an object's required references cannot be removed; mapping an object's required reference to Remove results in an error when trying to transport the object. For example, a job is required to have an agent, so if the job's agent is mapped to Remove you cannot transport the job.

Saving the Mapping of Data Objects

The mapping process can be time consuming depending upon the number of data objects you are working with. If you think you might copy the same data types between the same source and destination, you should preserve the mapping you have completed for the selected data objects. Creating a mapping file will save time and effort when repeating the copying process in the future. A *.map* extension is added when the mapping file is saved for easy identification.

Creating a Mapping File

To create a mapping file:

- Step 1** Complete the mapping of the data objects between the source and destination.
- Step 2** From the **File** menu, select **Save Map** to display the **Save TIDAL Scheduler Map File** dialog.

Step 3 Name and save the file to the desired directory location. Providing a qualified map file name will be helpful when searching for the file in the future.

An **Information** dialog confirms that the map file was created successfully.

The information within the map file can be examined by opening the file with a standard text editor program like Notepad.

Mapping Report

Transporter can display reports that list the data objects that were mapped in the current session.

To view a mapping report:

- From the **Reports** menu, select **Mapping Report**.

The **Scheduler Object Mapping Rules** report lists the mapping object type and its corresponding mapping rules that are described in a *.map* file. The report also provides the name of the map file and its source and destination. The Mapping Report is displayed in the default browser. See “[Sample Reports](#)” for an example. Alternatively, you can display the Mapping Report from the **Scheduler Object Mapping** dialog, by clicking the **Report** button.

Transporter can also list those data objects that it considers have invalid mapping because the data object was deleted or renamed in the destination master.

- From the **Reports** menu, select **Invalid Mappings**.

The **Invalid Mapped Object** report lists the mapping object type and its corresponding mapping rules that is considered invalid. The report also provides the name of the map file and its source and destination master. The Invalid Mapped Object report is displayed in the default browser. See “[Sample Reports](#)” for an example.

Mapping Timezones

TES can have an associated Timezone. For Transporter, this means that Timezones can be mapped from source to destination.

A Job can reference a Timezone which affects transporting of Jobs. Timezones are referenced via the **Scheduler Object Mapping** dialog.

The following example shows a source Job (with no Parent) with a referenced Timezone. There is a matching destination Timezone (AAA TIMEZONE).

The example below shows a Job (with a Parent, \JOB GOUP) with a referenced Timezone that is inherited from its Parent Group. There is a matching Timezone in the destination (TIMEZONE 01).

The following example shows a Job (with a Parent, \JOB GOUP) with a referenced Timezone that is NOT inherited from its Parent Group. There is a matching Timezone in the destination (TIMEZONE 01). Note that Timezone inheritance is independent of other inherited values (unlike Agent inheritance which is tied to AgentList and Runtime User inheritance).