



Transporting Data Objects

Overview

When object differences have been resolved using mapping rules, data objects can be easily transported. For Jobs, if some differences between the data objects still occur during the transport process, these differences can be resolved on-the-fly via the **Interactive Job Definition Mapping** dialog.



Warning

Changing the default Effective Date (current date) from the current date while transferring job groups may cause severe disruptions in the production schedule. Do not modify the Effective Date option to resolve this issue. Instead, use one of two methods to transfer a job group with a future effective date. Either disable the job group and then transfer it with the default Effective Date (current date) or select the Disable Copy and No Compile options in the Configuration Options dialog before transferring job groups. After the job group transfer is complete, enable the job group and schedule it as needed.

Transport Process

Once an object's referenced data objects have been optionally mapped between the designated source and destination, you can begin the transport process for that object. After selecting the data objects to be transported, you can start the transfer between the source and destination in one of four ways:

- Click the **Start Transfer** button on the toolbar.
- Select the **Start Transfer** option in the **Transport** main menu.
- For Jobs, drag and drop a single selected data object (or job group) from the Source side to job group on the Destination side.
- Press the **F9** key



Note

When dragging and dropping job/job groups to the Destination side, be careful that you do place them in the intended location. Verify that you are not inadvertently placing the items into the wrong job group.

If there are no problems during the data object transport process, an information dialog displays to notify you that the selected number of data objects was successfully transported to the destination.



Note

Re-reading object data by clicking the Read Data button on the toolbar or pressing the F5 key updates the screen with the current information.

Visual Cues

Transporter provides visual clues to provide information at-a-glance about the data objects displayed in the **Source** and **Destination**.

- The data objects in the **Source** display in blue to indicate that they exist in the destination (unless duplicate objects are filtered out). If the object exists in the destination but is disabled at the source, its text color will be teal.
- Source objects that do not exist in the destination are displayed in black text unless they are disabled, in which case their text is displayed in light gray.
- Newly transported objects in the **Destination** are displayed in blue text to indicate that they were copied during the current session. This is reset when object data is re-read.



Warning

If any error messages about connections are displayed while transferring data objects, disconnect from the source and destination by clicking the Disconnect button. Resolve the problem and reconnect to continue the data transfer.

Data types like calendars, variables, events, resources, actions, and job classes are simple to map since they do not have many object references. The owner of these type of data objects is usually is the only item that may need to be mapped.

Transporting Job Objects

Jobs are more complicated because of the multiple objects that a job can reference. While there are numerous types of data objects referenced in jobs, Transporter is primarily interested in the following data objects that are key to jobs in TES:

- **Owner** – The owner of the job or job group.
- **Agent** – The agent that will run the job.
- **Job Alias** – The alias name for the job.
- **Parent** – The parent of a child job.
- **Job Class** – The job class, if any, associated with the job or job group.
- **Calendar** – The calendar, if any, associated with the job or job group.
- **Agent List** – The agent list, if any, associated with the job or job group.
- **Runtime User** – The runtime user, if any, associated with the job or job group.
- **Variables** – The variables associated with a job.
- **Job Events** – The job events associated with the jobs.
- **Job Dependencies** – The job dependencies that exist between the jobs.
- **Variable Dependencies** – The variable dependencies that exist for the jobs.
- **File Dependencies** – The file dependencies that exist among the jobs.
- **Timezone** – The timezone defined among the jobs.
- **Virtual Resources** – The virtual resources defined among the jobs.
- **Custom Resources** – The custom resources defined among the jobs.

Discrepancies between the values of these data objects in each source and destination should be resolved by mapping an equivalent value in the destination to the data object in the source. However, if left unmapped, only a subset of these values are considered errors and failure to map these will prevent the job transport. Depending on the job type, these required fields are **Owner**, **Agent**, **Agent List**, and occasionally **Runtime User**.

Occasionally, when transporting jobs, a mapping error or conflict may occur. On those occasions, the **Interactive Job Definition Mapping** dialog is displayed to pinpoint and resolve the problem areas.

Interactive Job Definition Mapping Dialog

The **Interactive Job Definition Mapping** dialog displays during job transport operations unless the **Auto Add** configuration option was selected. If the **Auto Add** option is in affect, the **Interactive Job Definition Mapping** dialog is only displayed if a mapping error occurs while transporting the job.

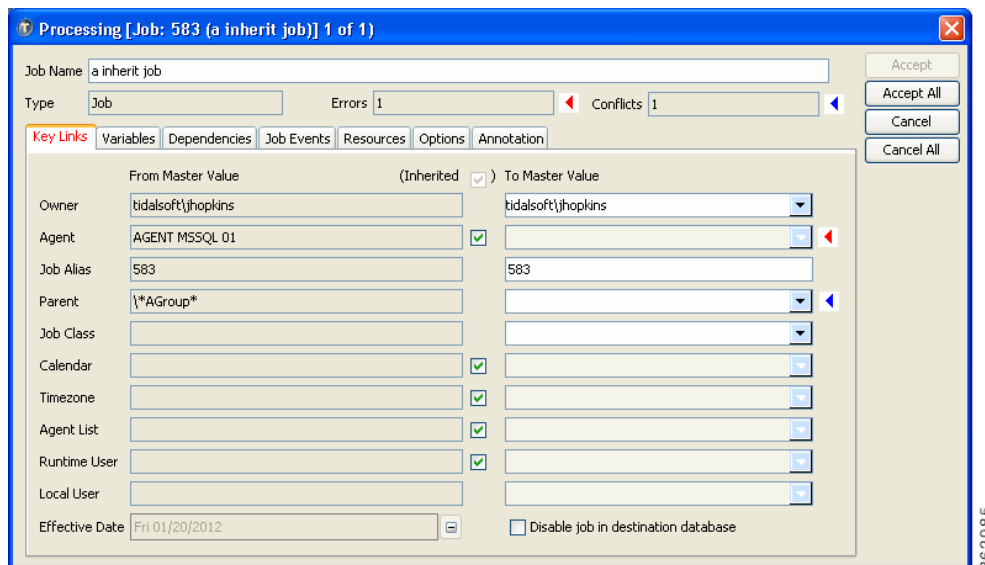


Figure 7-1 Interactive Job Definition Mapping Dialog, Key Links Tab

The title bar of the **Interactive Job Definition Mapping** dialog displays the current point of progress in the transport operation (1 of 1). The following text fields are displayed:

- **Job Name** – The name of the job being transported.
- **Type** – Indicates the job type (Job, Job Group, FTP job, etc.)
- **Errors** – Indicates the number of errors that occurred due to failed mappings between source and destination objects. Failed mapping for fields such as Owner, Agent or Agent List and sometimes Runtime Users (depending on job type) are considered errors. An error prevents a job from being transported.
- **Conflicts** – Indicates the number of conflicts that occurred due to failed mappings between source and destination objects. Conflicts indicate a job is missing non-critical data such as dependencies or events. While the job may still run with conflicts, the missing data may prevent the job from running as expected.

**Note**

You can use the **Delete** key to clear out non-required fields and if a field has conflicts, the conflict is also cleared.

There are several tabs in the **Interactive Job Definition Mapping** dialog:

- **Key Links** – The **Key Links** tab displays the basic attributes of a job. Three of the fields have required mappings, otherwise an error results. These three critical fields are the **Owner**, **Agent**, **Agent List**, and **Job Alias**.

**Note**

A unique job alias is auto-generated by Transporter and does not usually require user intervention.

Any one of these fields in the **To Master Value** column that do not map to values displayed in the **From Master Value** column displays a red arrow to highlight a mapping error.

When transporting a job with a parent, the **Key Links** tab displays an **Inherited** check box next to fields that may be inherited from its parent job. The fields are the **Agent**, **Calendar**, **Timezone**, **Agent List**, and **Runtime User** fields.

If the source job inherits attributes from its parent, then the check box will be selected next to the **Inherited** check box. If there is an attribute that is currently not inheriting, you can select the **Inherited** check box next to the attribute you want the copied job to inherit. However, if the job cannot inherit the attribute due to a failure mapping a corresponding value from the destination, then an error results preventing transport of the job. You must resolve this error to continue with the job transport.

In [Figure 7-1](#), the source job inherits its Agent, Calendar, and Runtime User from its group **AGroup**. A parent mapping conflict occurred because the parent **AGroup** does not exist and is not mapped to an alternate group at the destination. However, a parent group is not a required attribute of a job, which is why this failed mapping is considered a conflict. Because there is no parent selection, the inherited values that cannot be determined are blank in the **To Master** (Destination) side. The non-critical inherited values are displayed as conflicts in blue. The **Agent** field is displayed as an error in red because an agent or agent list is required for jobs. There are several options to resolve this error:

- A parent selection can be made as in the following figure. In this case, the selection of group *Test Group* resolved the Agent error because this group has an agent assignment of *AGENT MSSQL 01 (MSSql)*. This group also has a calendar and Runtime User.
- A second alternative is to select an agent by clearing the agent **Inherited** check box. This makes the destination **Agent** list available for selection of an agent value to resolve the agent error. Doing so automatically clears the **Runtime User** check box. You are left with only the Parent and Calendar conflicts. You can resolve these conflicts by selecting a parent group and calendar. To select a calendar, clear the **Calendar** inherited check box. This makes the destination Calendar list available for selection of a calendar

**Note**

If you have **Allow Conflicts** configured in the configuration option, you will be permitted to transport this job, even with conflicts.

If the job is transported with the Parent and Calendar conflict, the resulting job at the destination will not include a Parent nor a Calendar reference.

A third option is to make an Agent List selection. You can clear the **Agent List** inherited check box which enables the **Agent List** combo box selection, and then make a selection. This clears the agent error and allows transport of the job depending on resolution of any remaining conflicts.

Setting Inheritance

The following example includes a job that does not inherit from its parent.

When this job is transported, an error in the **Agent** field occurs because the source Agent *AGENT MSSQL 01* is non-existent and is not mapped to an alternate destination value. Because an agent or agent list is required for jobs, this is presented as an error. This error can be resolved by making an agent or agent list selection.

Since the job includes a parent group, if the parent group has an Agent or Agent List assignment, the error can also be resolved by selecting the **Agent** or **Agent List Inherited** check box.

After selecting the **Agent Inherited** check box, the error is resolved. The corresponding destination group has an Agent List assignment rather than an Agent assignment, as does its corresponding source group. In this case, selecting the **Agent Inherited** check box selects the Agent List value that is associated with the destination parent group. Note that the **Runtime User** field is automatically set to inherit and its value selected to match the destination parent's assigned Runtime User. This is because Agent/Agent List inheritance also assumes Runtime User inheritance.

Selecting the **Agent List Inherited** check box has the same result. When this job is transported, it is created under the group indicated and its Agent/Agent List value and Runtime User values inherit from this group.



Note

If the destination parent group had no Agent or Agent List assignment, checking the **Agent** or **Agent List Inherited** check boxes would not resolve the error.

- **Variables** – The **Variables** tab displays the variables associated with the job named in the **Job Name** field. These variables are arranged in two columns, **From Master Variable** and **To Master Variable**. At the bottom of the **Variables** tab is a **Show Usage** button that displays how the variables are used in the job. Unmapped variables are displayed as conflicts since jobs do not usually require variables.



Note

You can use the **Delete** key to clear out non-required fields. This is the same as selecting <<**Remove**>> from the destination list

- **Dependencies** – The **Dependencies** tab displays the dependencies associated with the job named in the **Job Name** field. These dependencies are divided into three types of dependencies: job dependencies, variable dependencies and file/JES dependencies. Each type of dependency is arranged in two columns, **From Master Dependency** and **To Master Variable**. Unmapped dependencies are displayed as conflicts since jobs do not usually require dependencies.



Note

You can use the **Delete** key to clear out non required fields. This is the same as selecting <<**Remove**>> from the destination list

**Note**

In the **To Master Agent** list, use the <<**Remove**>> option to remove the corresponding File or JES dependency. This option is not used to remove an Agent.

- **Job Events** – This tab displays the job events associated with the job named in the **Job Name** field. These job events are arranged in two columns, **From Master Event** and **To Master Event**.
- **Resources** – This tab displays the job resources associated with the job named in the **Job Name** field. These resources are arranged in two columns, **From Master Resource** and **To Master Resource**.

**Note**

You can use the **Delete** key to clear out non-required fields. This is the same as selecting <<**Remove**>> from the destination list.

- **Options** – This tab allows you to set the default handling of job output.
- **Annotation** – When an object is transported, a description of the transport operation is annotated to the **Description** field of the object, if **Annotate** is in effect. This option provides for a customizable annotation message to the job description during transport.

The **Interactive Job Definition Mapping** dialog displays the following buttons:

- **Accept** – Accepts the changes you have made, saves the job, and displays the next job to be mapped, if any. When there are no more jobs to process, the dialog is closed. If a mapping error is not resolved, the **Accept** button is unavailable and you can only cancel the job and go to the next job for processing. The dialog updates with data for each job in process.
- **Accept All** – Accepts all selected jobs for transport without displaying the mapping dialog for the job. Jobs with errors will be skipped. Jobs with only conflicts will be accepted only if **Allow Conflicts** is enabled. Use this option only if you are confident that all items were properly mapped earlier. Selecting the **Accept All** button displays the message “Only jobs without conflicts or errors will be accepted automatically. Proceed?”
- **Cancel** – Closes the displayed job’s mapping dialog without saving any changes to that job and moves on to the next job, if any.
- **Cancel All** – Closes the **Job Mapping** dialog and cancels processing for jobs that have not yet been processed during the current transport operation. Jobs that were accepted earlier during the session will still be transported.

Resolving Mapping Errors and Conflicts For Jobs

The **Interactive Job Definition Mapping** dialog provides a means to resolve any differences between the source and destination on-the-fly.

Some data object differences are more critical than others. Mapping differences are divided into two types called *Errors* and *Conflicts*. Transporter provides visual clues to help the user quickly pinpoint any mapping problems between source and destination. Colored text in the **Interactive Job Definition Mapping** dialog highlights the tab and field where mapping differences exist.

Errors

Errors denote more crucial differences between data object. An error occurs when a vital component of a job in the source does not exist or is not mapped to an equivalent object in the destination. Errors result from missing values in the corresponding data fields that are key to a job. These data fields are the **Owner**, **Agent**, **Agent List**, and **Job Alias**. You can locate the field with the error by looking for the tab with red text. On that tab, a red arrow will point to each field containing an error. A job with a mapping error cannot be transported until the error is resolved.

From the destination list for the **Error** field, select a value in the destination that should be mapped to the source value. The number in the **Errors** field will decrease each time an error is resolved. The job cannot be successfully transported to the destination until all errors are resolved.

Conflicts

Conflicts are the less critical difference between data objects. A conflict occurs when a job object referenced by the source does not have a corresponding match in the destination. Unlike errors, these attributes are not critical. You can locate the field with the conflict by looking for the tab with the blue text. On that tab, one or more blue arrows will point to the field(s) causing the conflict. Conflicts do not prevent the job from being copied, but the conflicts result in a loss of data if not resolved (e.g., it may be missing dependencies or associated events, etc.)

From the destination list corresponding to the **Conflict** field, select a value in the destination that should be mapped to the source value. The number in the **Conflicts** field will decrease each time a conflict is resolved.

Transporting Within the Same Database

When dealing with the same database, the source and destination job cannot be the same. It must be mapped to a new parent if there is an existing parent or to a new parent if none exists.



Note

Transporting within the same database only applies to Jobs, all other tabs will be disabled.

The transported job is then created under the parent group specified, with the original job undisturbed. For example, suppose you have a job group called Sales, which includes jobs relating to sales that were originally created under a group called Dev while the jobs are under development. Now, suppose you wish to migrate this job to production. You can transport the Sales group, but it must first be mapped to a new group.

Sample Transport Within the Same Database

In the example described above, the following job structure exists where **job B** depends on **job A** and **job C** depends on **job B**:

`\DEVSALES\job A, \DEVSALES\job B, \DEVSALES\job C`

To transport group `\DEVSALES` and its child jobs **job A** through **job C**:

Step 1 Create a mapping rule for Group DEV. In this example, DEV is mapped to PROD.

**Note**

In the drop-down menu for group \DEV, you will not be provided an option to map \DEV to itself.

Step 2

Transport **SALES** and its child jobs. Notice that its Parent Job is mapped to **PROD** as per the mapping rule above. **SALES** will be created under parent **PROD** once transported.

Job A is transported under group \PROD\SALES.

Job B is transported under group \PROD\SALES.

Job B is transported with its job dependency on job \PROD\SALES\Job A.

Job C is transported under group \PROD\SALES and similarly along with its job dependency on job \PROD\SALES\Job b.

Because this is the first time that the DEV\SALES group was transported, Transporter creates a new group called **SALES** under the mapped group **PROD**.

A default annotation is included in the **Notes** tab that indicates the job was created.

After the initial transport, subsequent transport of group \DEV\SALES results in an update to existing \PROD\SALES.

A default annotation is included in the Job notes that indicates the job was updated.

If \PROD\SALES is transported, there will be an error because **PROD** had not been mapped. To resolve this, you can either create a new mapping rule for group **PROD** via the **Interactive Job Definition Mapping** dialog or you can interactively select a new group from the drop list. \PROD will not be available for selection.
