

My Datastage Notes

Pages

- Home
- Different Versions
- All Datastage Stages
- Configuration File
- Sequential_Stage
- Dataset
- Transformer Stage
- Sort Stage
- Aggregator_Stage
- Join Stage
- Lookup_Stage
- Merge_Stage
- Filter_Stage
- Copy Stage
- Funnel_Stage
- Column Generator
- Surrogate_Key_Stage
- SCD
- Pivot_Enterprise_Stage
- Sequence_Activities
- Datastage Study Material/Interview Questions
- Datastage Errors and Resolution
- Datastage Scenarios and solutions
- Unix Shell Scripting
- SQL/Database
- Datawarehousing Concepts

Blog Archive

- ▼ 2014 (46)
 - September (40)

Sunday, May 11, 2014

Local Containers and Shared Container

A **container**, as its name indicates, is used to group stages and links. Containers help simplify and modularize server job designs and allow you to replacing complex areas of the diagram with a single container stage. For example, if you have a lookup that is used by multiple jobs, you can put the jobs and links that generate the lookup into a share container and use it to different jobs. In a way, you can look at it like a procedure or function in the programming term.

Containers are linked to other stages or containers in the job by input and output stages.

Two types of container:

1. **Local containers.** These are created within a job and are only accessible by that job. A local container is edited in a tabbed page of the job's Diagram window. Local containers can be used in server jobs or parallel jobs.

2. **Shared containers.** These are created separately and are stored in the Repository in the same way that jobs are. There are two types of shared container:

(1.) Server shared container. Used in server jobs (can also be used in parallel jobs).

(2.) Parallel shared container. Used in parallel jobs. You can also include server shared containers in parallel jobs as a way of incorporating server job functionality into a parallel stage (for example, you could use one to make a server plug-in stage available to a parallel job).

For here, I only tested the Server Jobs so I only put notes on the server jobs now. Parallel Jobs works differently. Will note on it as a separate topic.

1 Local Container

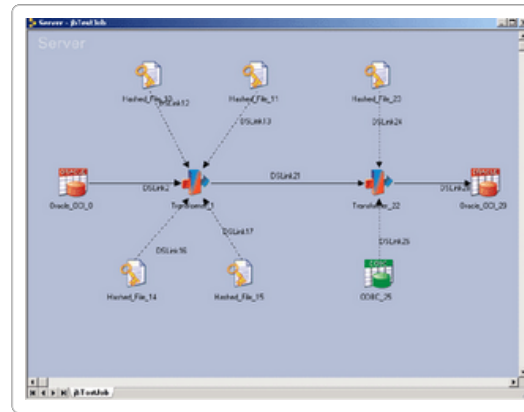
The main purpose of using a DataStage local container is to simplify a complex design

- ▼ May (4)
 - Resource Estimation tool
 - Local Containers and Shared Container
 - SCD Type 2
 - Dimension and Facts
- February (2)
- 2013 (39)

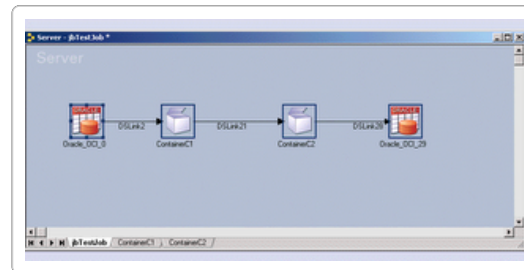
visually to make it easier to understand in the Diagram window. If the DataStage job has lots of stages and links, it may be easier to create additional containers to describe a particular sequence of steps.

To create a local container, from an existing job design, do the following:

(1.) Press the Shift key and using the mouse to click on the stages that you want to put into the local container.



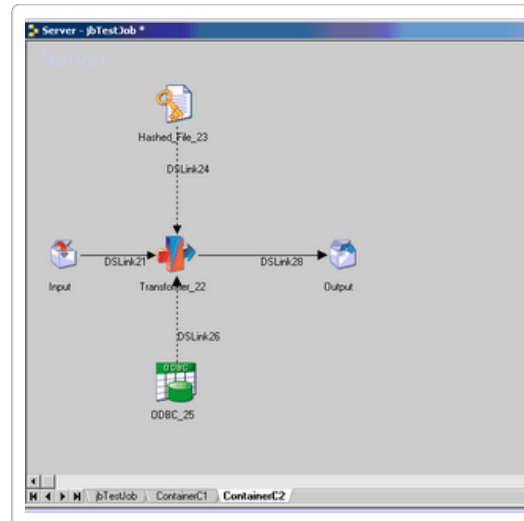
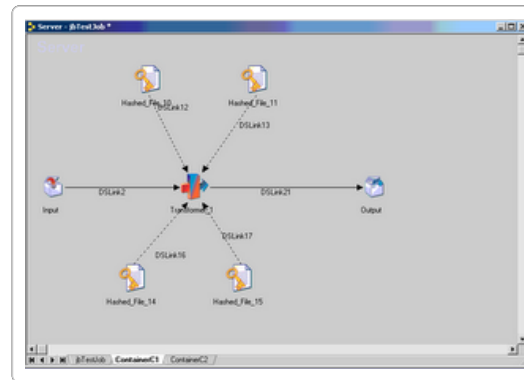
(2.) From the Menu bar, select Edit ► Construct Container ► Local.



The group is replaced by a Local Container stage in the Diagram window. A new tab appears in the Diagram window containing the contents of the new Local Container stage. You are warned if any link naming conflicts occur when the container is constructed. The new container is opened and focus shifts onto its tab.

You can rename, move, and delete a container stage in the same way as any other stage in your job design.

To view or modify a local container, just double-click container stage in the Diagram window. You can edit the stages and links in a container in the same way you do for a job.



To create a empty container to which you can add stages and links, drag and drop the Container icon in the General group on the tool palette onto the Diagram window.

A Container stage is added to the Diagram window, double-click on the stage to open it, and add stages and links to the container the same way you do for a job.

1.1 Using Input and Output Stages in a local container

Input and output stages are used to represent the stages in the main job to which the container connects.

-- If you construct a local container from an existing group of stages and links, the input and output stages are automatically added. The link between the input or output stage and the stage in the container has the same name as the link in the main job Diagram window.

In the example above, the input link is the link connects to the container from the main job's

Oracle_OCI stage (oracle_oci_0). The output link is the link that connects to the second container from the first container.

-- If you create a new container, it will place the input and output stages in the container without any link. You must add stages to the container Diagram window between the input and output stages. Link the stages together and edit the link names to match the ones in the main Diagram window.

You can have any number of links into and out from of a local container, all of the link names inside the container must match the link names into and out of it in the job. Once a connection is made, editing meta data on either side of the container edits the meta data on the connected stage in the job.

2 Share Container

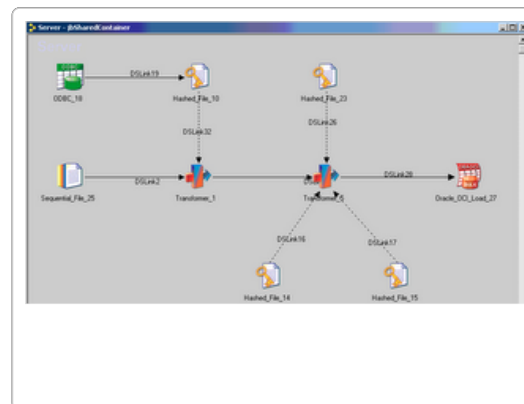
Shared containers also help you to simplify your design but, unlike local containers, they are reusable by other jobs. You can use shared containers to make common job components available throughout the project.

Shared containers comprise groups of stages and links and are stored in the Repository like DataStage jobs. When you insert a shared container into a job, DataStage places an instance of that container into the design. When you compile the job containing an instance of a shared container, the code for the container is included in the compiled job. You can use the DataStage debugger on instances of shared containers used within jobs.

You can create a shared container from scratch, or place a set of existing stages and links within a shared container.

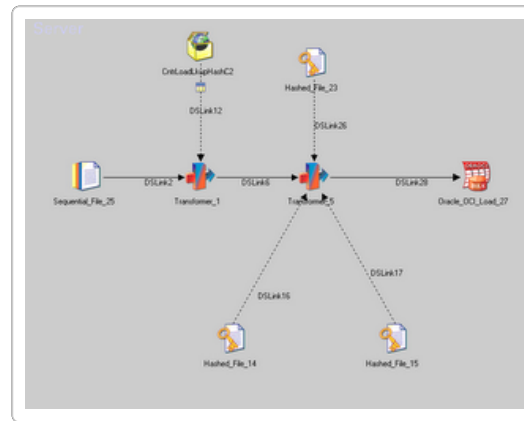
2.1 Create a shared container from an existing job design

(1.) Press Shift and click the other stages and links you want to add to the container.



(2.) From the Menu bar, select Edit > Construct Container > Shared. You will be prompted for a name for the container by the Create New dialog box. The group is replaced by a

Shared Container stage of the appropriate type with the specified name in the Diagram window.



Any parameters occurring in the components are copied to the shared container as container parameters. The instance created has all its parameters assigned to corresponding job parameters.

(3.) Modify or View a Shared Container

Select File ->Open from the Menu bar and select to Shared Container that you want to open. You can also highlight the Shared Container and use right mouse click and select property.

2.2 User A Shared Container

(1.) Dragging the Shared Container icon from the Shared Container branch in the Repository window to the job's Diagram window.

(2.) Update the Input and Output tabs.

-- Map to Container Link.

Choose the link within the shared container to which the incoming job link will be mapped. Changing the link triggers a validation process, and you will be warned if the meta data does not match and are offered the option of reconciling the meta data as described below.

-- Columns page

Columns page shows the meta data defined for the job stage link in a standard grid. You can use the Reconcile option on the Load Shared Containers button to overwrite meta data on the job stage link with the container link meta data in the same way as described for the Validate option.

Posted by manohar at 9:10 AM

No comments:

[Post a Comment](#)

[Newer Post](#)

[Home](#)

[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)

Manohar. Simple template. Powered by [Blogger](#).