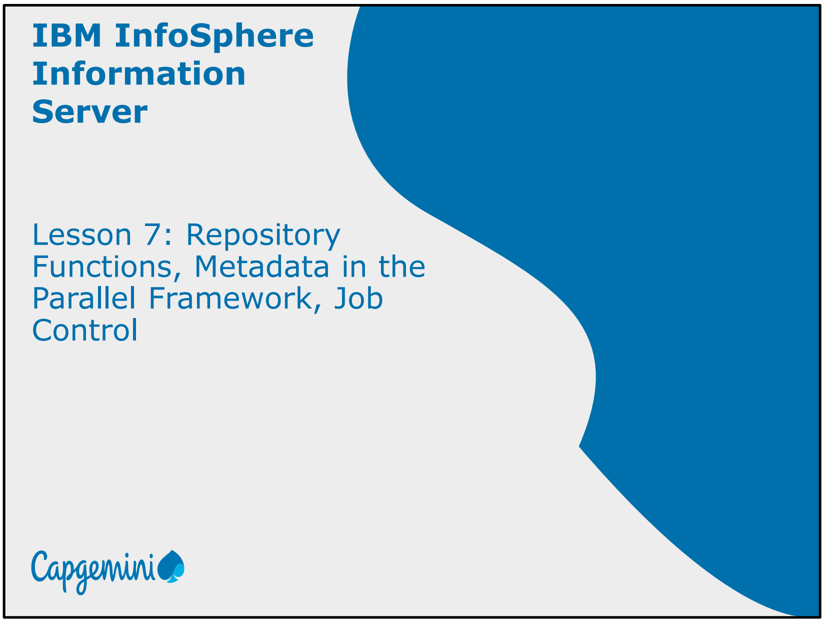


IBM InfoSphere Information Server

Lesson 7: Repository Functions, Metadata in the Parallel Framework, Job Control

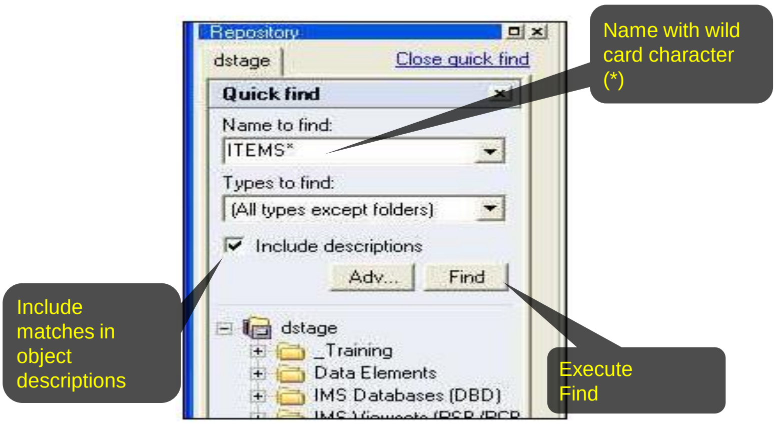
Lesson Objectives

- After completing this unit, you should be able to:
- Perform a simple Find
- Perform an Advanced Find
- Perform an impact analysis
- Compare the differences between two Table Definitions
- Compare the differences between two jobs

IBM InfoSphere Information Server

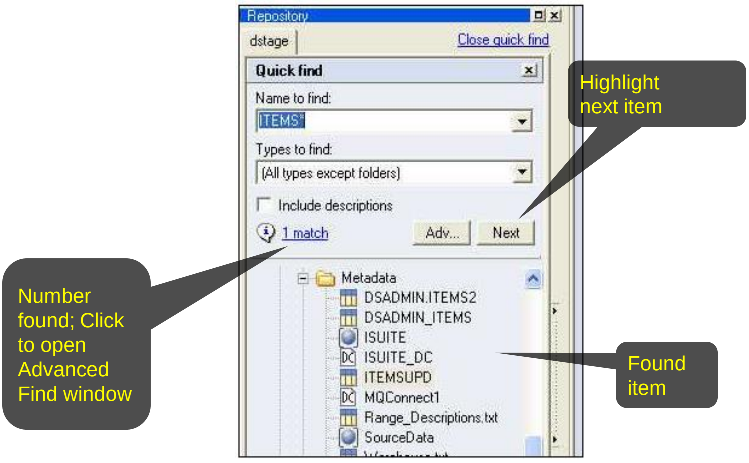
7.1 : Searching the Repository

Quick Find

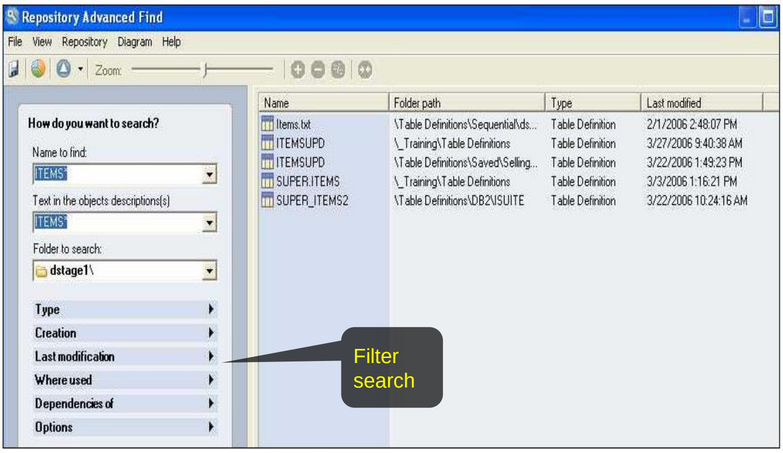


If the Include descriptions box is checked, the text in Short descriptions and Long descriptions will be searched.

Found Results



Advanced Find Window



How do you want to search?

Name to find: ITEMS*

Text in the objects descriptions(s): ITEMS*

Folder to search: dstage1\

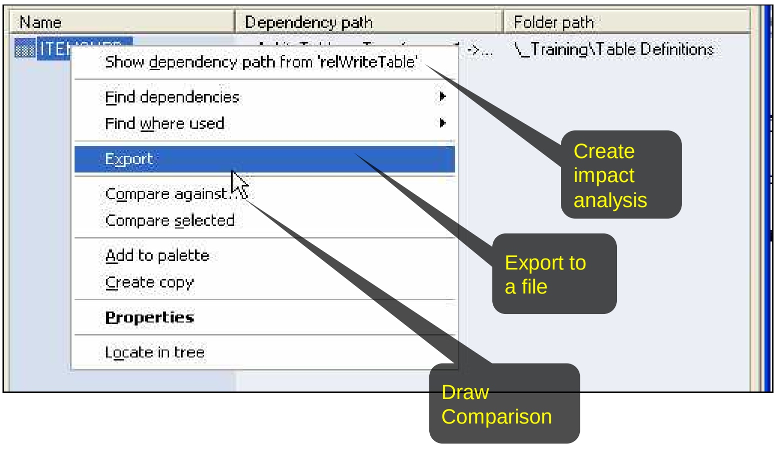
Type
Creation
Last modification
Where used
Dependencies of
Options

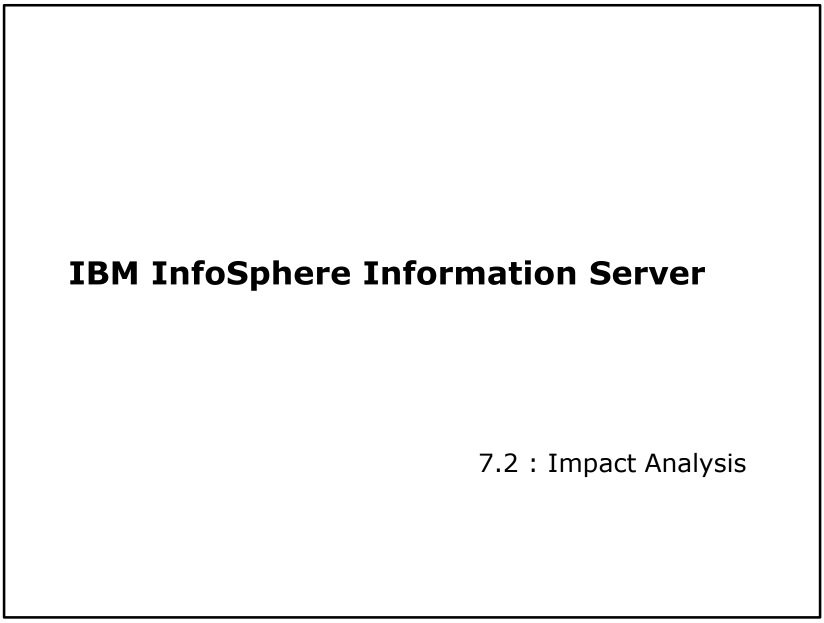
Name	Folder path	Type	Last modified
Items.txt	\Table Definitions\Sequential\ds...	Table Definition	2/1/2006 2:48:07 PM
ITEMSUPD	_Training\Table Definitions	Table Definition	3/27/2006 9:40:38 AM
ITEMSUPD	\Table Definitions\Saved\Selling...	Table Definition	3/22/2006 1:49:23 PM
SUPER.ITEMS	_Training\Table Definitions	Table Definition	3/3/2006 1:16:21 PM
SUPER_ITEMS2	\Table Definitions\DB2\ISUITE	Table Definition	3/22/2006 10:24:16 AM

Advanced Find Filtering Options

- Type: type of object
  - Job, Table Definition, etc.
- Creation: range of dates
  - E.g., Up to a week ago
- Last modification: range of dates
  - E.g., Up to a week ago
- Where used: objects that use specified objects
  - E.g., a job that uses a specified Table Definition
- Dependencies of: objects that are dependencies of objects
  - E.g., a Table Definition that is referenced in a specified job
- Options
  - Case sensitivity
  - Search within last result set

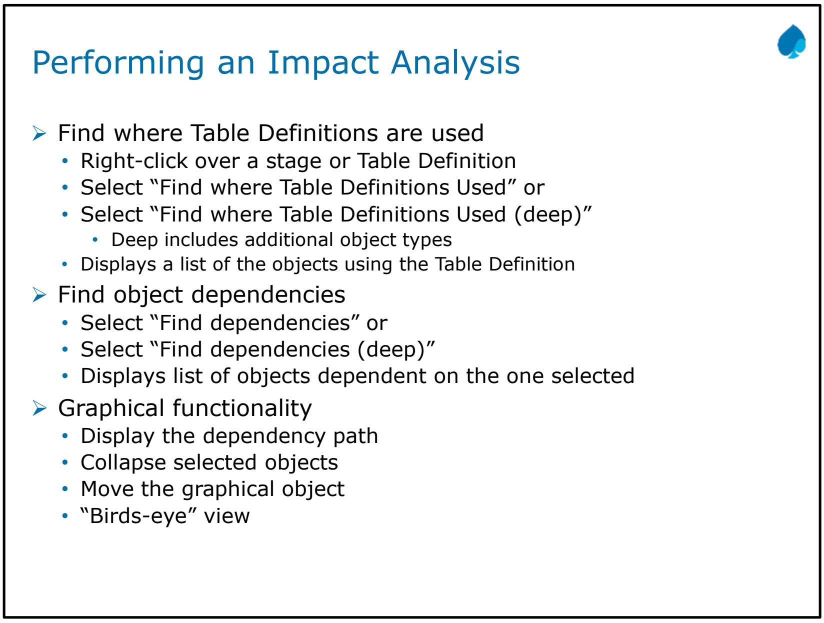
Using the Found Results





IBM InfoSphere Information Server

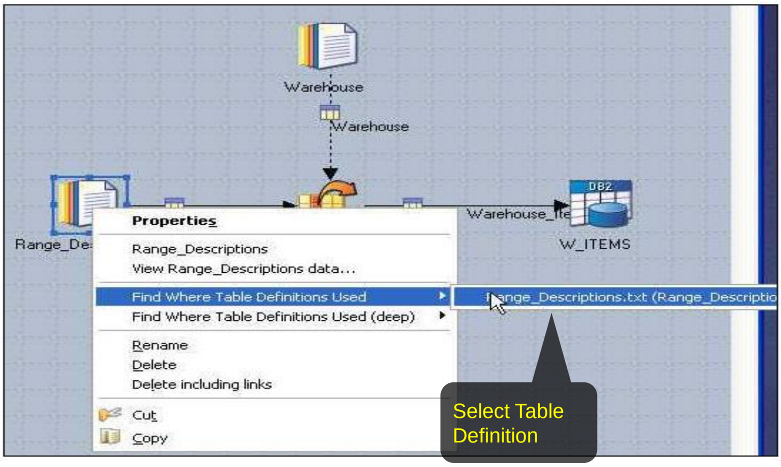
7.2 : Impact Analysis



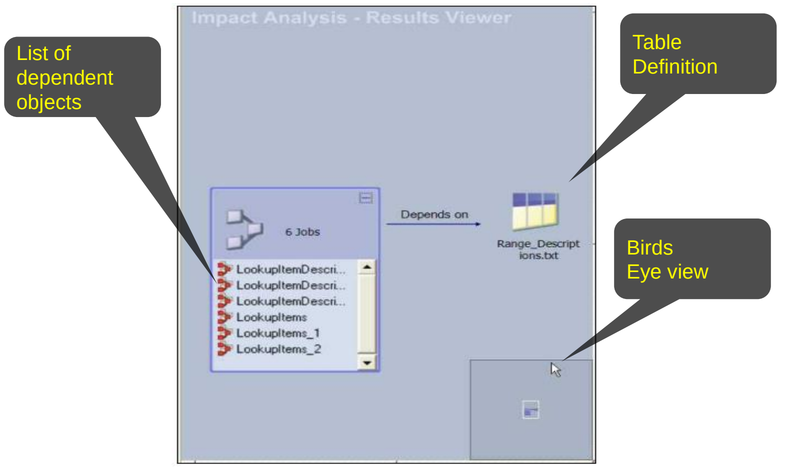
Performing an Impact Analysis

- Find where Table Definitions are used
  - Right-click over a stage or Table Definition
  - Select "Find where Table Definitions Used" or
  - Select "Find where Table Definitions Used (deep)"
    - Deep includes additional object types
  - Displays a list of the objects using the Table Definition
- Find object dependencies
  - Select "Find dependencies" or
  - Select "Find dependencies (deep)"
  - Displays list of objects dependent on the one selected
- Graphical functionality
  - Display the dependency path
  - Collapse selected objects
  - Move the graphical object
  - "Birds-eye" view

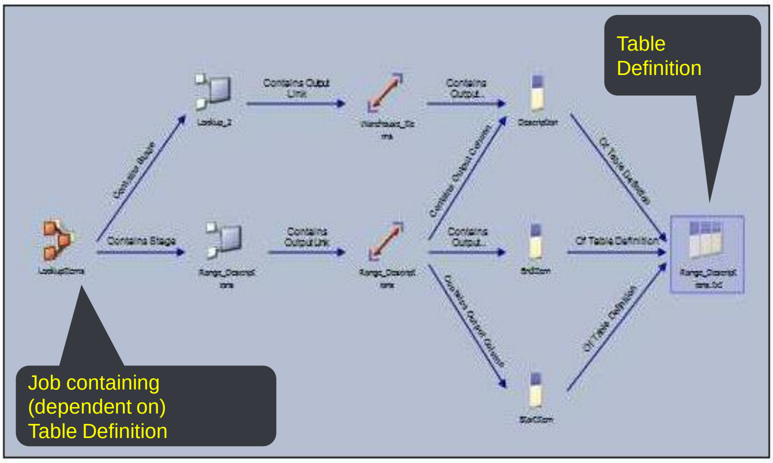
Initiating an Impact Analysis from a Stage


Warehouse
Warehouse
Range_De
Warehouse_Ite
DB2
W_ITEMS
Properties
Range_Descriptions
View Range_Descriptions data...
Find Where Table Definitions Used
Find Where Table Definitions Used (deep)
Rename
Delete
Delete including links
Cut
Copy
Range_Descriptions.txt (Range_Descriptio
Select Table Definition

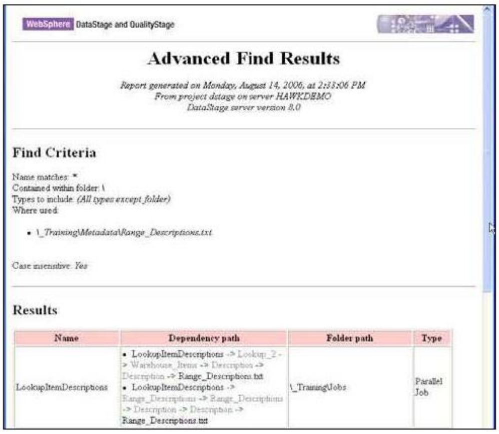
Displaying the Dependencies Graphically



Displaying the Dependency Path



Generating an HTML Report


WebSphere DataStage and QualityStage

Advanced Find Results

Report generated on Monday, August 14, 2006, at 2:33:06 PM
From project dstage on server HAWKDEMO
DataStage server version 8.0

Find Criteria

Name matches: *
Contained within folder: \
Types to include: *(All types except folder)*
Where used:

- *_Training\Metadata\Range_Descriptions.txt*

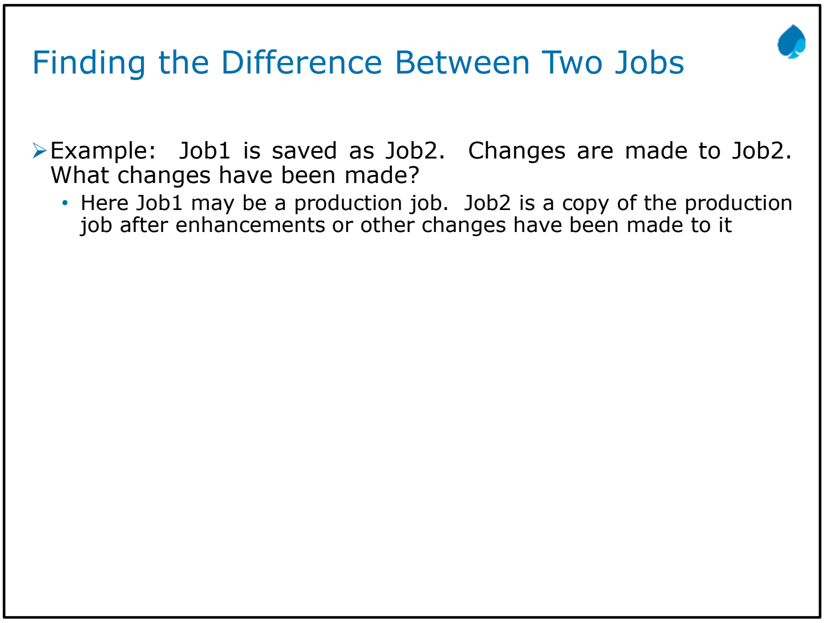
Case insensitive: *Yes*

Results

Name	Dependency path	Folder path	Type
LookupItemDescriptions	• LookupItemDescriptions -> Lookup_2 -> Warehouse_Items -> Description -> Description -> Range_Descriptions.txt • LookupItemDescriptions -> Range_Descriptions -> Range_Descriptions -> Description -> Description -> Range_Descriptions.txt	_Training\Jobs	Parallel Job

IBM InfoSphere Information Server

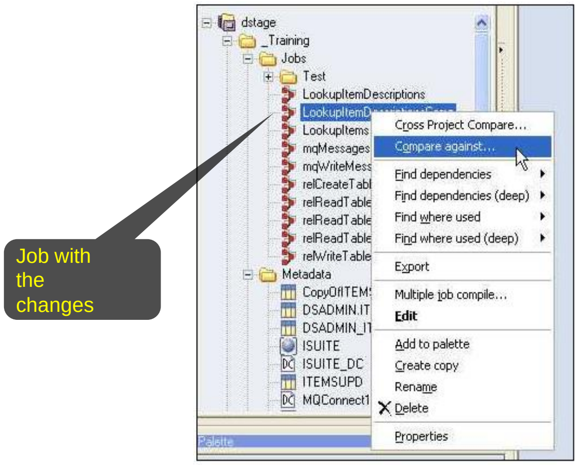
7.3 : Job and Table Difference Reports



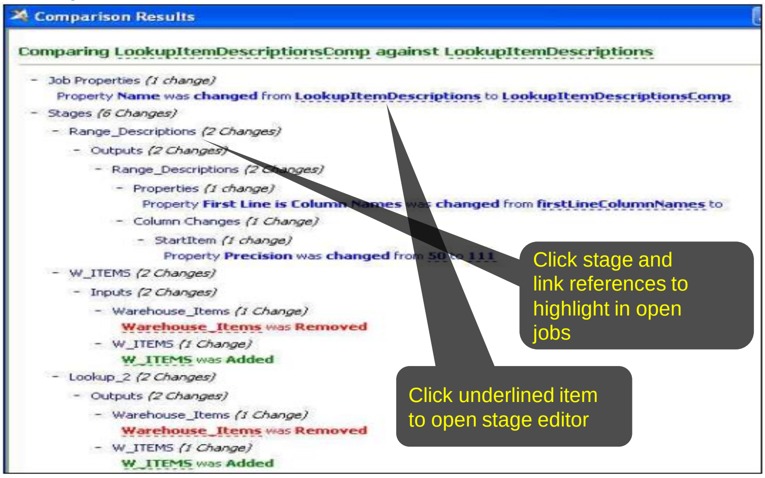
Finding the Difference Between Two Jobs

- Example: Job1 is saved as Job2. Changes are made to Job2. What changes have been made?
  - Here Job1 may be a production job. Job2 is a copy of the production job after enhancements or other changes have been made to it

Initiating the Comparison



Comparison Results

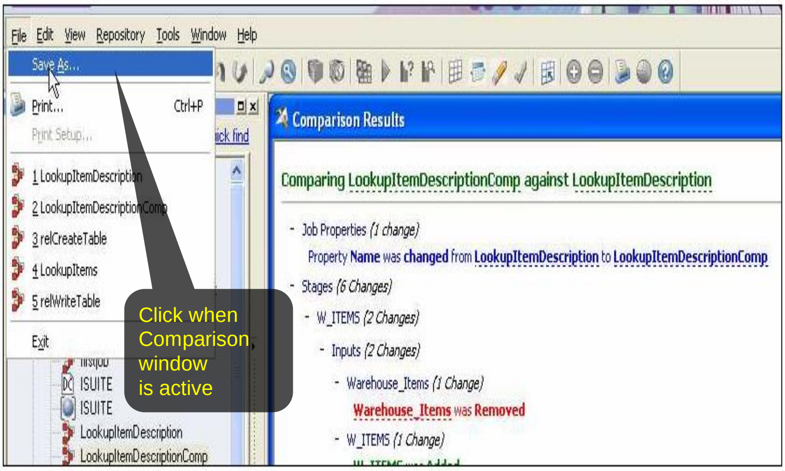

Comparison Results

Comparing LookupItemDescriptionsComp against LookupItemDescriptions

- Job Properties *(1 change)*
  - Property **Name** was **changed** from **LookupItemDescriptions** to **LookupItemDescriptionsComp**
- Stages *(6 Changes)*
  - Range_Descriptions *(2 Changes)*
    - Outputs *(2 Changes)*
      - Range_Descriptions *(2 Changes)*
        - Properties *(1 change)*
          - Property **First Line is Column Names** was **changed** from **firstLineColumnNames** to
        - Column Changes *(1 Change)*
          - StartItem *(1 change)*
            - Property **Precision** was **changed** from **50** to **111**
  - W_ITEMS *(2 Changes)*
    - Inputs *(2 Changes)*
      - Warehouse_Items *(1 Change)*
        - **Warehouse_Items** was **Removed**
      - W_ITEMS *(1 Change)*
        - **W_ITEMS** was **Added**
  - Lookup_2 *(2 Changes)*
    - Outputs *(2 Changes)*
      - Warehouse_Items *(1 Change)*
        - **Warehouse_Items** was **Removed**
      - W_ITEMS *(1 Change)*
        - **W_ITEMS** was **Added**

Click stage and link references to highlight in open jobs
Click underlined item to open stage editor

Saving to an HTML File

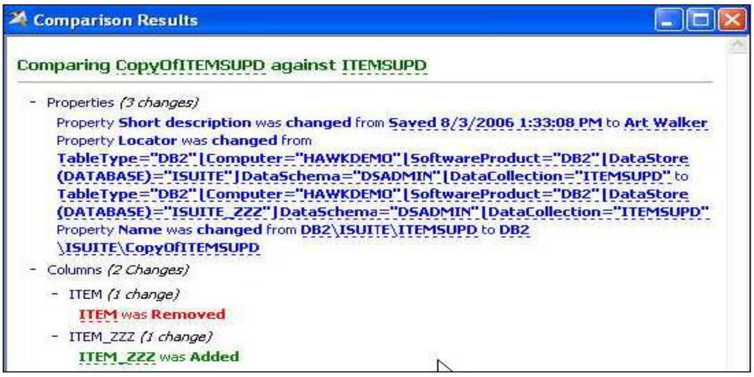


Comparing LookupItemDescriptionComp against LookupItemDescription

- Job Properties *(1 change)*
  - Property **Name** was **changed** from **LookupItemDescription** to **LookupItemDescriptionComp**
- Stages *(6 Changes)*
  - W_ITEMS *(2 Changes)*
    - Inputs *(2 Changes)*
      - Warehouse_Items *(1 Change)*
        - **Warehouse_Items** was **Removed**
      - W_ITEMS *(1 Change)*

Comparing Table Definitions

- Same procedure as when comparing jobs

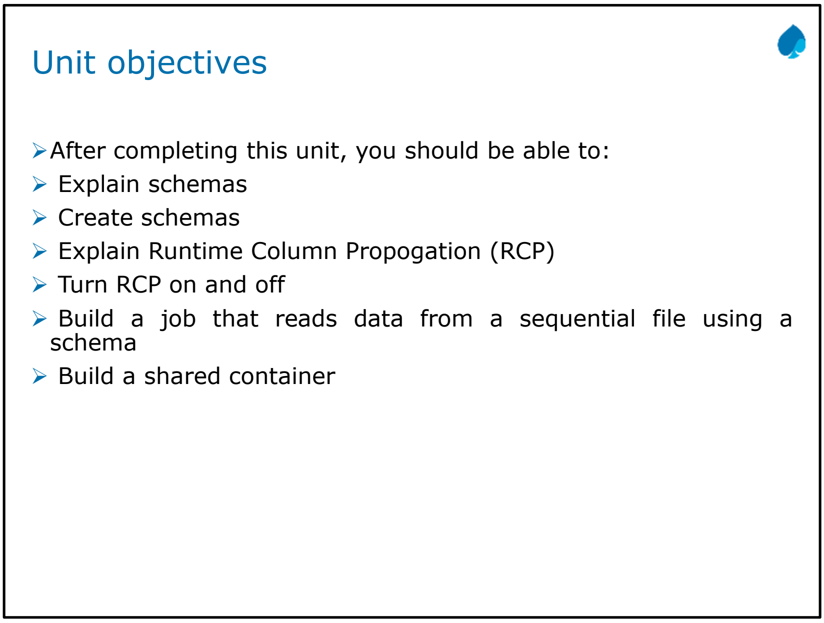


Comparing CopyOfITEMSUPD against ITEMSUPD

- Properties *(3 changes)*
  - Property **Short description** was **changed** from **Saved 8/3/2006 1:33:08 PM** to **Art Walker**
  - Property **Locator** was **changed** from **TableType="DB2"|Computer="HAWKDEMO"|SoftwareProduct="DB2"|DataStore (DATABASE)="ISUITE"|DataSchema="DSADMIN"|DataCollection="ITEMSUPD"** to **TableType="DB2"|Computer="HAWKDEMO"|SoftwareProduct="DB2"|DataStore (DATABASE)="ISUITE_ZZZ"|DataSchema="DSADMIN"|DataCollection="ITEMSUPD"**
  - Property **Name** was **changed** from **DB2\ISUITE\ITEMSUPD** to **DB2 \ISUITE\CopyOfITEMSUPD**
- Columns *(2 Changes)*
  - ITEM *(1 change)*
    - ITEM was **Removed**
  - ITEM_ZZZ *(1 change)*
    - ITEM_ZZZ was **Added**

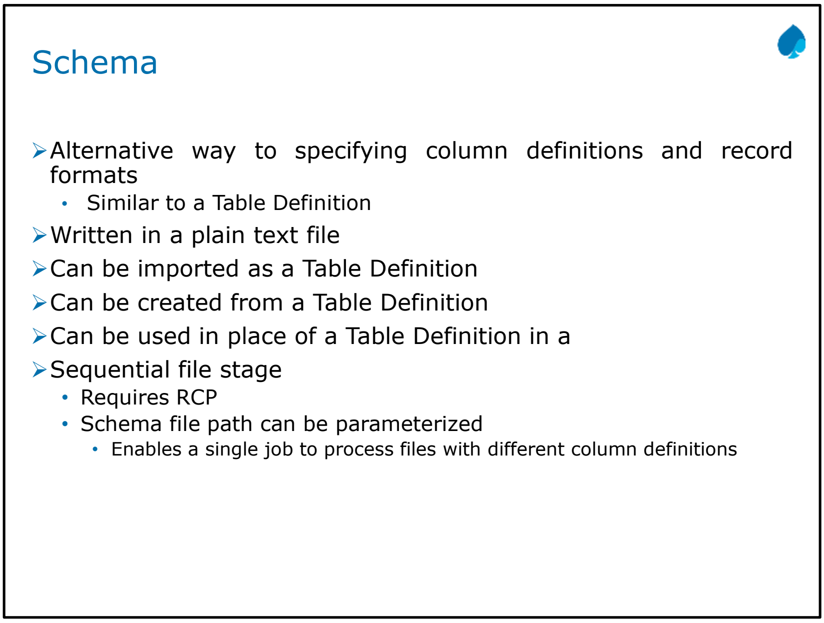
IBM InfoSphere Information Server

7.4 : Metadata in the Parallel Framework



Unit objectives

- After completing this unit, you should be able to:
- Explain schemas
- Create schemas
- Explain Runtime Column Propogation (RCP)
- Turn RCP on and off
- Build a job that reads data from a sequential file using a schema
- Build a shared container



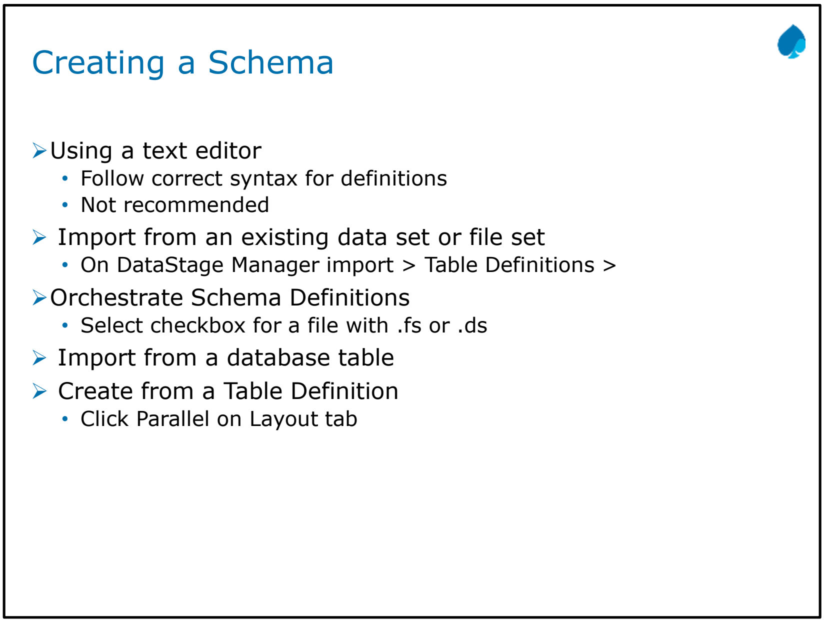
Schema

- Alternative way to specifying column definitions and record formats
  - Similar to a Table Definition
- Written in a plain text file
- Can be imported as a Table Definition
- Can be created from a Table Definition
- Can be used in place of a Table Definition in a
- Sequential file stage
  - Requires RCP
  - Schema file path can be parameterized
    - Enables a single job to process files with different column definitions

The format of each line describing a column is:
column_name:[nullability]datatype;

column_name.
This is the name that identifies the column. Names must start with a letter or an underscore (_), and can contain only alphanumeric or underscore characters. The name is not case sensitive. The name can be of any length.
nullability.
You can optionally specify whether a column is allowed to contain a null value, or whether this would be viewed as invalid. If the column can be null, insert the word ’nullable’. By default columns are not nullable.

You can also include ’nullable’ at record level to specify that all columns are nullable, then override the setting for individual columns by specifying ‘not nullable’. For example:
record nullable (
name:not
nullable string[255];
value1:int32;
date:date
)
datatype.
This is the data type of the column.

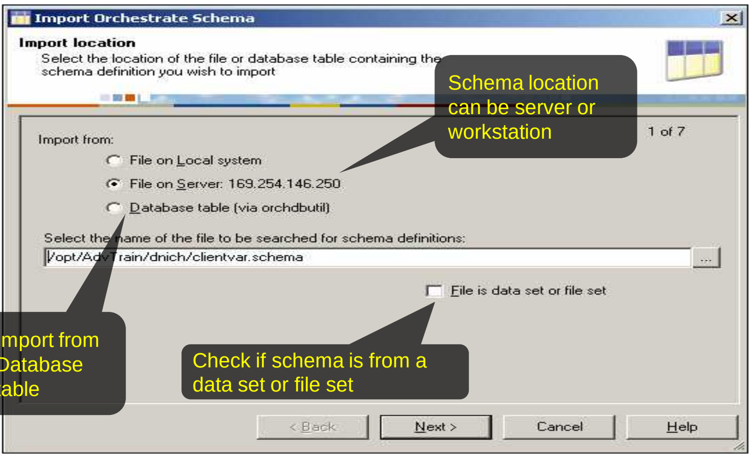


Creating a Schema

- Using a text editor
  - Follow correct syntax for definitions
  - Not recommended
- Import from an existing data set or file set
  - On DataStage Manager import > Table Definitions >
- Orchestrate Schema Definitions
  - Select checkbox for a file with .fs or .ds
- Import from a database table
- Create from a Table Definition
  - Click Parallel on Layout tab

Another good way of capturing a schema is to set $OSH_PRINT_SCHEMAS and copy entries from the DataStage Director log.

Importing a Schema



Import Orchestrate Schema

Import location

Select the location of the file or database table containing the schema definition you wish to import

1 of 7

Import from:

- ( ) File on Local system
- (●) File on Server: 169.254.146.250
- ( ) Database table (via orchdbutil)

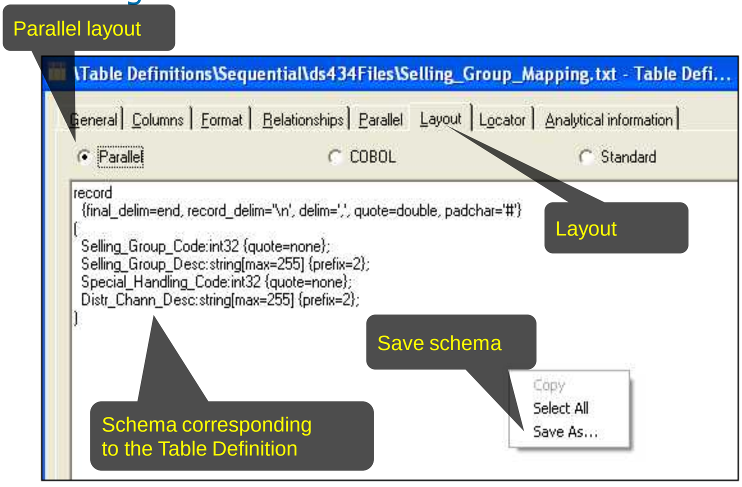
Select the name of the file to be searched for schema definitions:

/opt/AdvTrain/dnich/clientvar.schema

[ ] File is data set or file set

...
< Back
Next >
Cancel
Help
Schema location can be server or workstation
Import from Database table
Check if schema is from a data set or file set

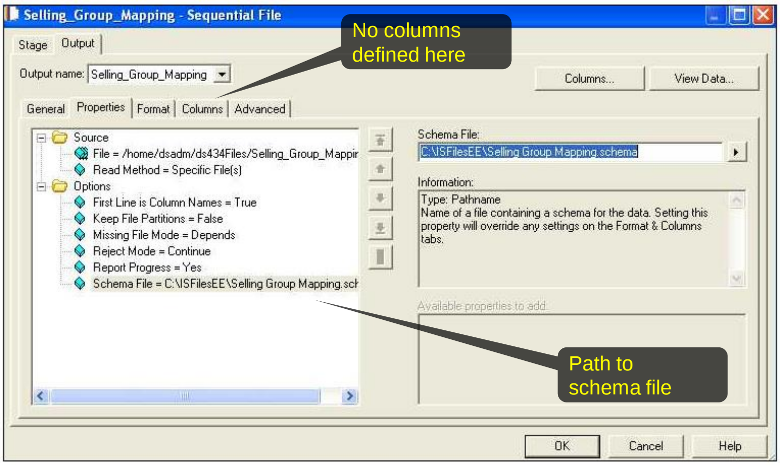
Creating a Schema From a Table Definition



[✓ Parallel] COBOL Standard

```
record
 {final_delim=end, record_delim='\n', delim=',', quote=double, padchar='#'}
(
 Selling_Group_Code:int32 {quote=none};
 Selling_Group_Desc:string[max=255] {prefix=2};
 Special_Handling_Code:int32 {quote=none};
 Distr_Chann_Desc:string[max=255] {prefix=2};
)
```

Reading a Sequential File Using a Schema



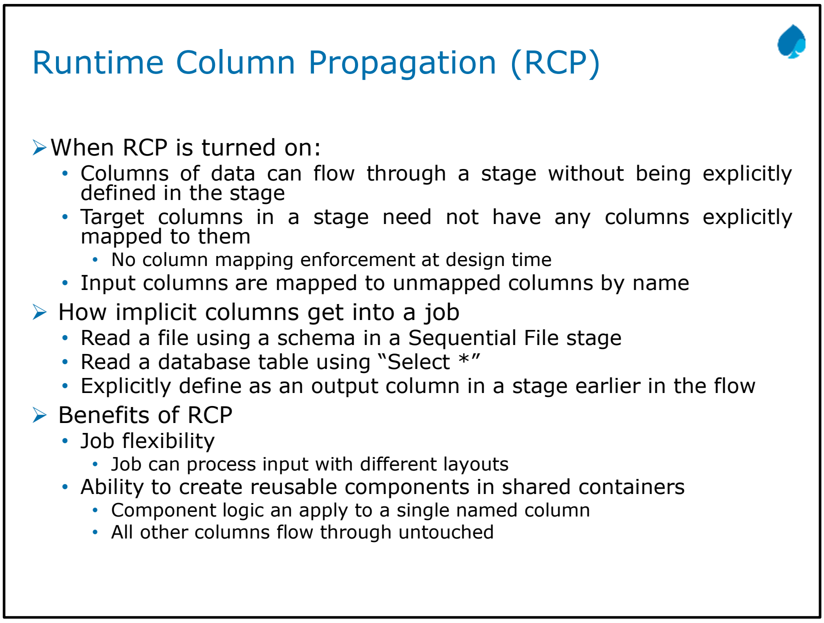
- Source
  - File = /home/dsadm/ds434Files/Selling_Group_Mappir
  - Read Method = Specific File(s)
- Options
  - First Line is Column Names = True
  - Keep File Partitions = False
  - Missing File Mode = Depends
  - Reject Mode = Continue
  - Report Progress = Yes
  - Schema File = C:\ISFilesEE\Selling Group Mapping.sch

Schema File: C:\ISFilesEE\Selling Group Mapping.schema

Information:
Type: Pathname
Name of a file containing a schema for the data. Setting this property will override any settings on the Format & Columns tabs.

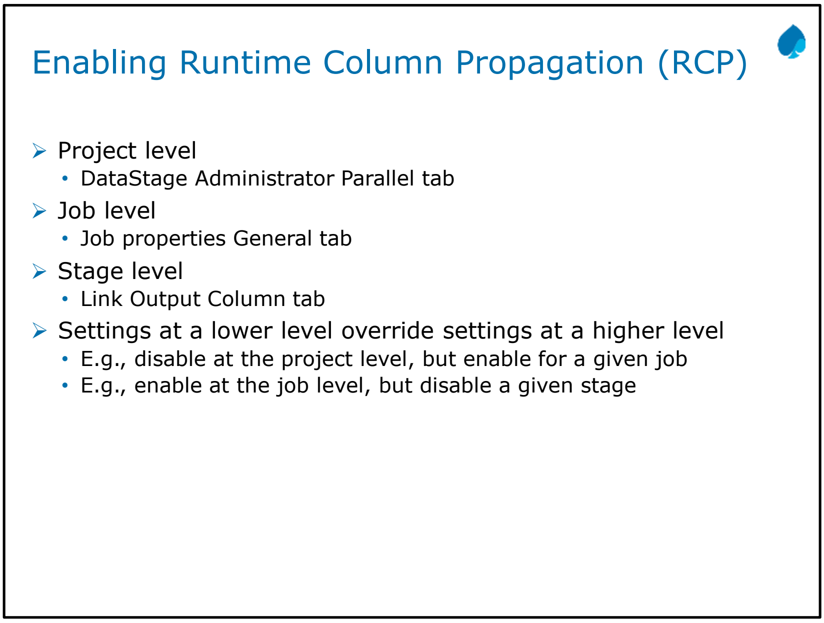
Available properties to add:

Schemas can only be used when Runtime Column Propogation is turned on in the stage. This is discussed later in this module.



Runtime Column Propagation (RCP)

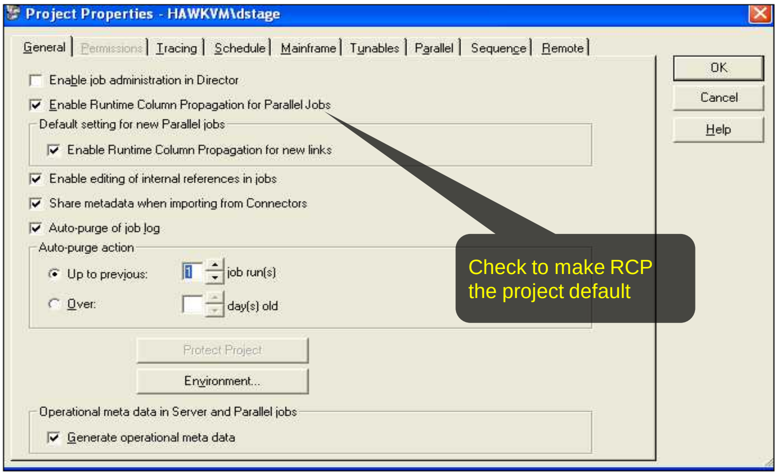
- When RCP is turned on:
  - Columns of data can flow through a stage without being explicitly defined in the stage
  - Target columns in a stage need not have any columns explicitly mapped to them
    - No column mapping enforcement at design time
  - Input columns are mapped to unmapped columns by name
- How implicit columns get into a job
  - Read a file using a schema in a Sequential File stage
  - Read a database table using “Select *”
  - Explicitly define as an output column in a stage earlier in the flow
- Benefits of RCP
  - Job flexibility
    - Job can process input with different layouts
  - Ability to create reusable components in shared containers
    - Component logic an apply to a single named column
    - All other columns flow through untouched



Enabling Runtime Column Propagation (RCP)

- Project level
  - DataStage Administrator Parallel tab
- Job level
  - Job properties General tab
- Stage level
  - Link Output Column tab
- Settings at a lower level override settings at a higher level
  - E.g., disable at the project level, but enable for a given job
  - E.g., enable at the job level, but disable a given stage

Enabling RCP at Project Level


Project Properties - HAWKVM\dstage

General | Permissions | Tracing | Schedule | Mainframe | Tunables | Parallel | Sequence | Remote

- ☐ Enable job administration in Director
- ☑ Enable Runtime Column Propagation for Parallel Jobs

Default setting for new Parallel jobs

- ☑ Enable Runtime Column Propagation for new links

- ☑ Enable editing of internal references in jobs
- ☑ Share metadata when importing from Connectors
- ☑ Auto-purge of job log

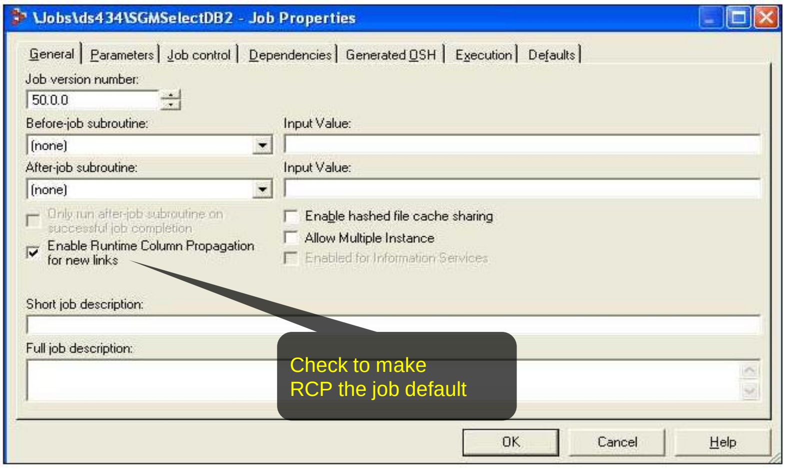
Auto-purge action

- [● Up to previous:] 1 job run(s)
- ○ Over: day(s) old

Operational meta data in Server and Parallel jobs

- ☑ Generate operational meta data

Enabling RCP at Job Level


\Jobs\ds434\SGMSelectDB2 - Job Properties

General | Parameters | Job control | Dependencies | Generated OSH | Execution | Defaults

Job version number: 50.0.0

Before-job subroutine: (none) Input Value:

After-job subroutine: (none) Input Value:

☐ Only run after-job subroutine on successful job completion

☐ Enable hashed file cache sharing

☐ Allow Multiple Instance

☑ Enable Runtime Column Propagation for new links

☐ Enabled for Information Services

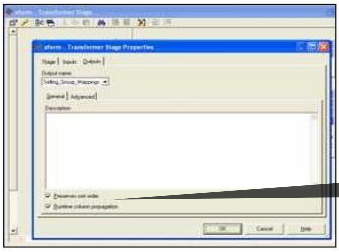
Short job description:

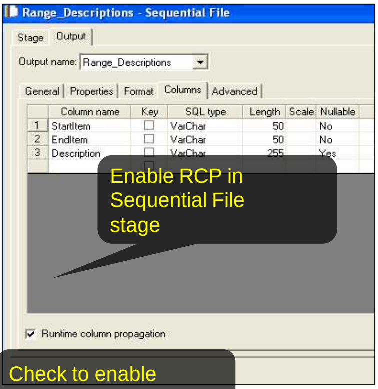
Full job description:

OK | Cancel | Help

Enabling RCP at Stage Level

- Sequential File stage
  - Output Columns tab
- Transformer
  - Open Stage Properties
  - Stage Properties Output tab


xform - Transformer Stage Properties
Stage
Inputs
Outputs
Output name:
General
Advanced
Description
Runtime column propagation
OK
Cancel
Help
Check to enable RCP in Transformer


Range_Descriptions - Sequential File
Stage
Output
General
Properties
Format
Columns
Advanced
Enable RCP in Sequential File stage

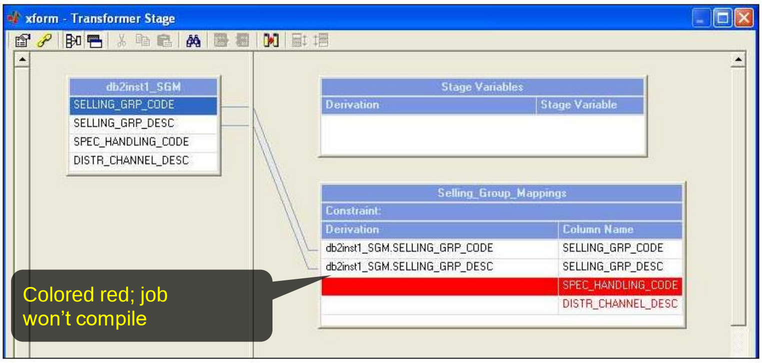
Output name: Range_Descriptions

	Column name	Key	SQL type	Length	Scale	Nullable
1	StartItem	☐	VarChar	50		No
2	EndItem	☐	VarChar	50		No
3	Description	☐	VarChar	255		Yes

☑ Runtime column propagation

When RCP is Disabled

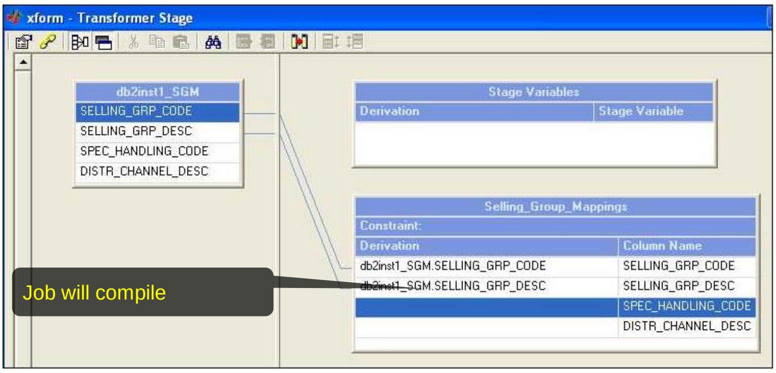
- DataStage Designer enforces Stage Input to Output column mappings.



Modify operators can add or change columns in a data flow.

When RCP is Enabled

- DataStage does not enforce mapping rules
- Runtime error if no incoming columns match unmapped target column names



IBM InfoSphere Information Server

7.5 : Shared Containers

Shared Containers

- Encapsulate job design components into a stored container
- Provide reusable job design components
- Example
  - Apply stored Transformer business logic

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

Types of Containers

➢Two Types:

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.

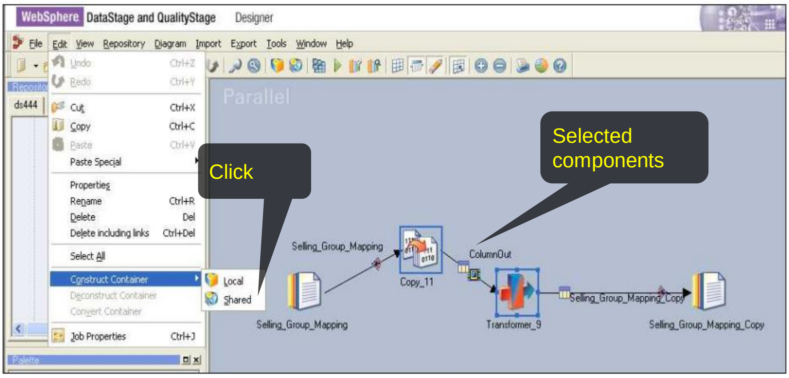
Shared Containers

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

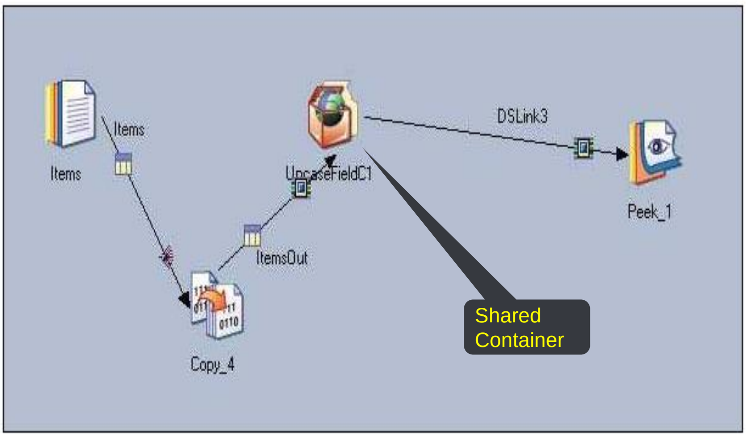
  Parallel shared container. Used in parallel jobs. You can also include server shared containers in parallel jobs as a way of incorporating server job functionality into a parallel stage (for example, you could use one to make a server plug-in stage available to a parallel job).

Creating a Shared Container

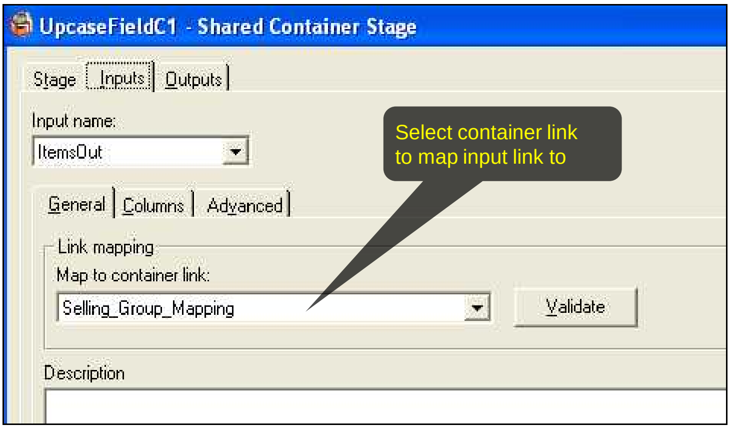
- Select stages from an existing job
- Click Edit>Construct



Using a Shared Container in a Job



Mapping Input / Output Links to the Container



UpcaseFieldC1 - Shared Container Stage

Input name: ItemsOut

Link mapping

Map to container link: Selling_Group_Mapping

Select container link to map input link to

Unit summary

Having completed this unit, you should be able to:

- Perform a simple Find
- Perform an Advanced Find
- Perform an impact analysis
- Compare the differences between two Table Definitions
- Compare the differences between two jobs
- Explain schemas
- Create schemas
- Explain Runtime Column Propagation (RCP)
- Turn RCP on and off
- Build a shared container

Q&A

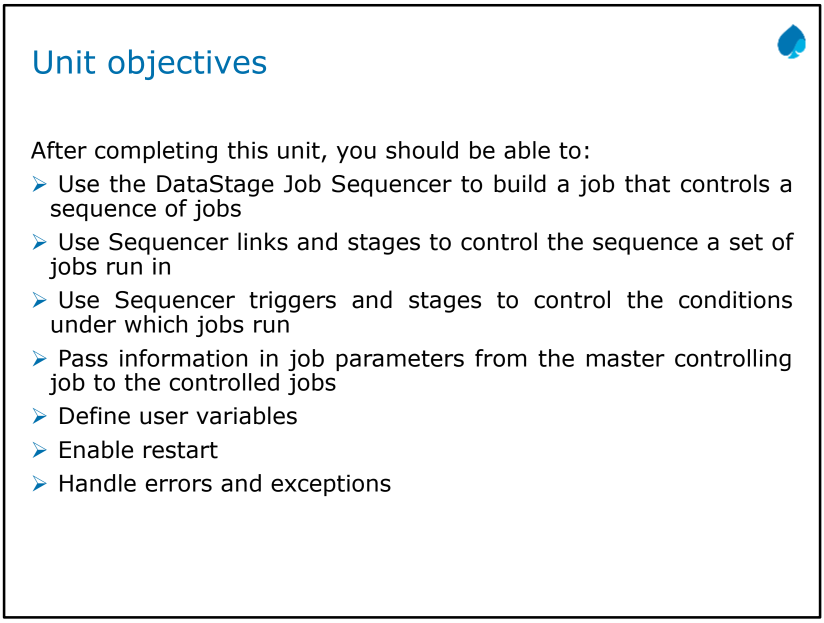
- You can compare the differences between what two kinds of objects?
- What “wild card” characters can be used in a Find?
- You have a job whose name begins with “abc”. You can’t remember the rest of the name or where the job is located. What would be the fastest way to export the job to a file?
- Name three filters you can use in a Advanced Find?
- What are two benefits of RCP?
- What can you use to encapsulate stages and links in a job to make them reusable?

Q&A

- Jobs. Table Definitions.
- Asterisk (*). It stands for any zero or more characters.
- Do a Find for objects matching "abc*". Filter by type job. Locate the job in the result set, click the right mouse button over it, and then click Export.
- Type of object, creation date range, last modified date range, where used, dependencies of, other options including case sensitivity and search within last result set.
- Job flexibility. Ability to create reusable components.
- Shared containers

IBM InfoSphere Information Server

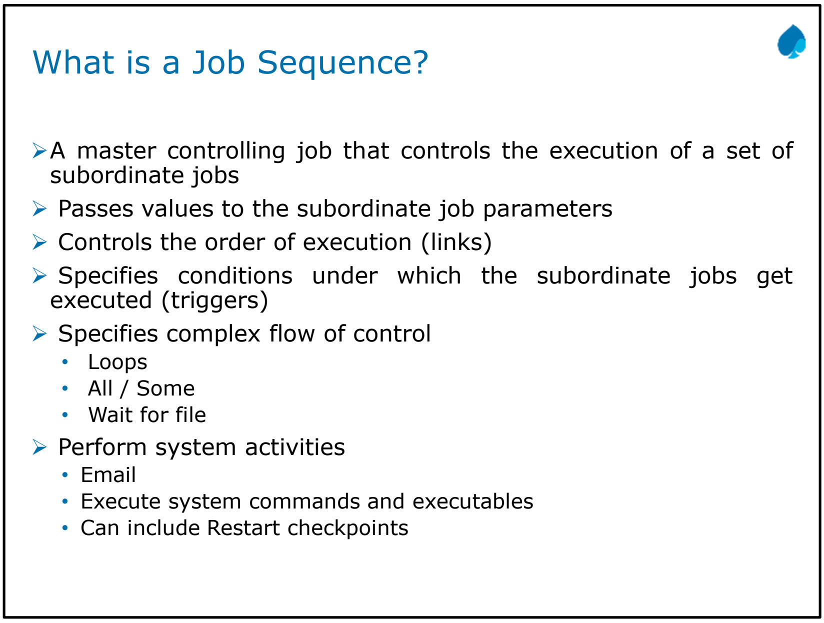
7.6 : Job Control



Unit objectives

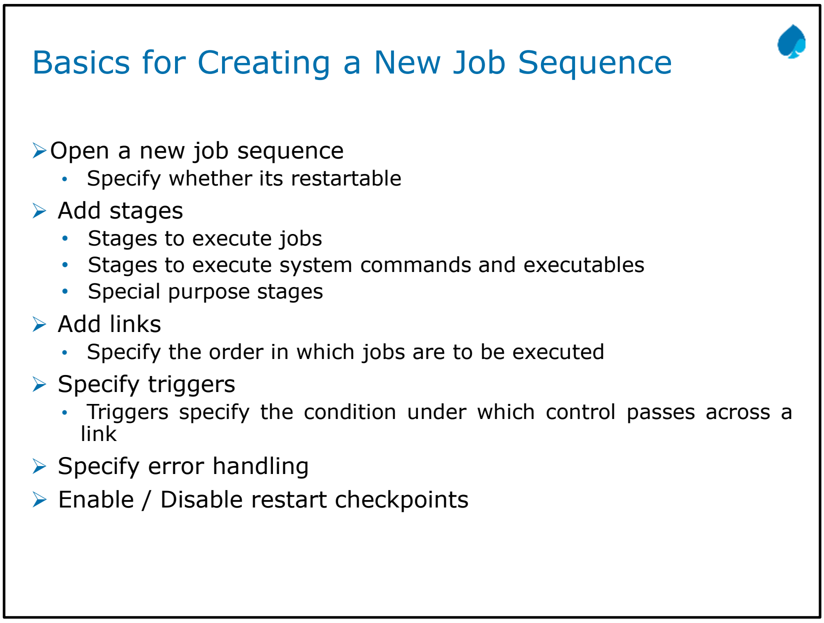
After completing this unit, you should be able to:

- Use the DataStage Job Sequencer to build a job that controls a sequence of jobs
- Use Sequencer links and stages to control the sequence a set of jobs run in
- Use Sequencer triggers and stages to control the conditions under which jobs run
- Pass information in job parameters from the master controlling job to the controlled jobs
- Define user variables
- Enable restart
- Handle errors and exceptions



What is a Job Sequence?

- A master controlling job that controls the execution of a set of subordinate jobs
- Passes values to the subordinate job parameters
- Controls the order of execution (links)
- Specifies conditions under which the subordinate jobs get executed (triggers)
- Specifies complex flow of control
  - Loops
  - All / Some
  - Wait for file
- Perform system activities
  - Email
  - Execute system commands and executables
  - Can include Restart checkpoints

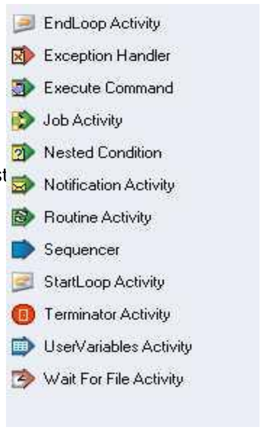


Basics for Creating a New Job Sequence

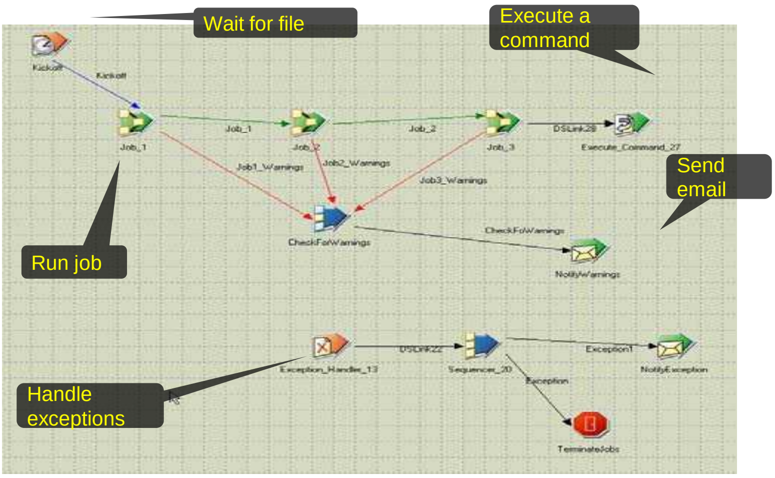
- Open a new job sequence
  - Specify whether its restartable
- Add stages
  - Stages to execute jobs
  - Stages to execute system commands and executables
  - Special purpose stages
- Add links
  - Specify the order in which jobs are to be executed
- Specify triggers
  - Triggers specify the condition under which control passes across a link
- Specify error handling
- Enable / Disable restart checkpoints

Job Sequencer Stages

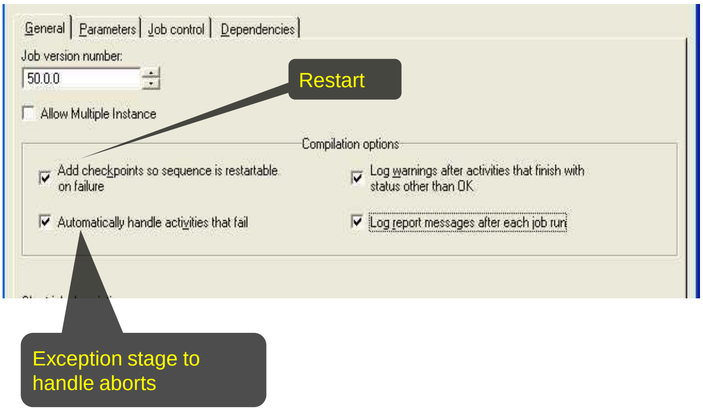
- Run stages
  - Job Activity: Run a job
  - Execute Command: Run a system command
  - Notification Activity: Send an email
- Flow control stages
  - Sequencer: Go if All / Some
  - Wait for File: Go when file exists / doesn't exist
  - Start Loop / End Loop
  - Nested Condition: Go if condition satisfied
- Error handling
  - Exception Handler
  - Terminator
- Variables
  - User Variables



Example


Wait for file
Execute a command
Kickoff
Kickoff
Job_1
Job_1
Job_2
Job_2
Job_3
Job_3
DSLink28
Execute_Command_27
Job1_Warnings
Job2_Warnings
Job3_Warnings
Send email
CheckForWarnings
CheckForWarnings
NotifyWarnings
Run job
Exception_Handler_13
DSLink22
Sequencer_20
Exception1
NotifyException
Exception
TerminateJobs
Handle exceptions

Sequence Properties


General | Parameters | Job control | Dependencies

Job version number: 50.0.0

☐ Allow Multiple Instance

Compilation options

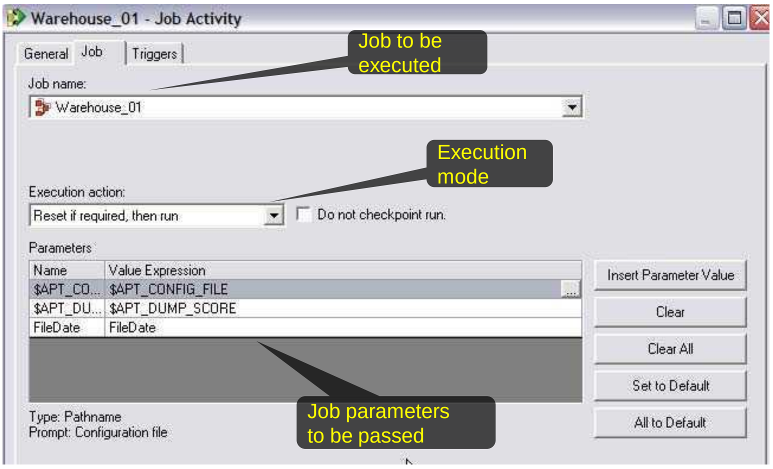
☑ Add checkpoints so sequence is restartable on failure

☑ Log warnings after activities that finish with status other than OK

☑ Automatically handle activities that fail

☑ Log report messages after each job run

Job Activity Stage Properties



Job name: Warehouse_01

Execution action: Reset if required, then run

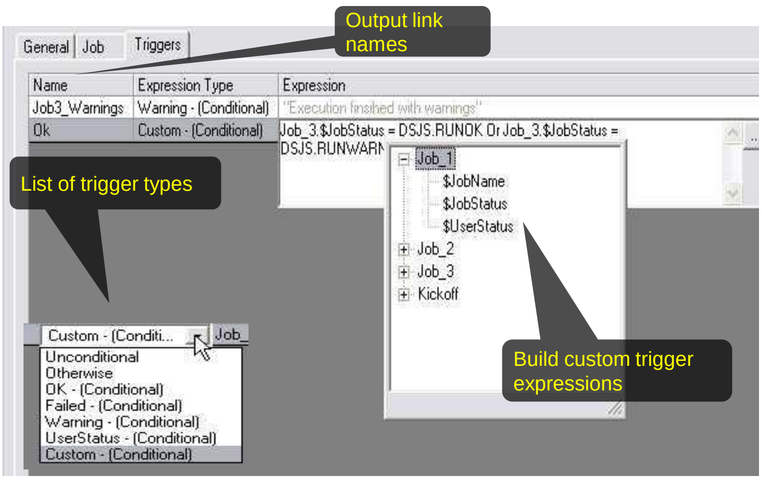
Do not checkpoint run.

Parameters

Name	Value Expression
$APT_CO...	$APT_CONFIG_FILE
$APT_DU...	$APT_DUMP_SCORE
FileDate	FileDate

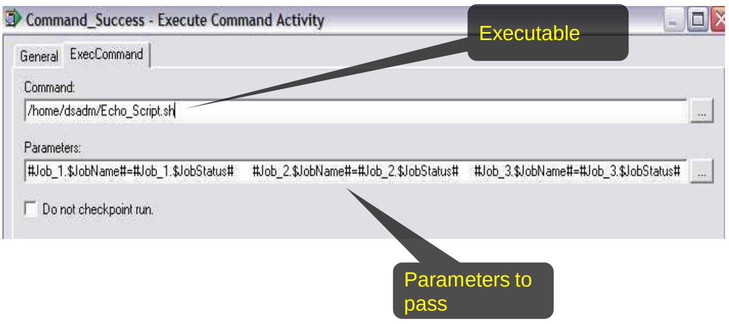
Type: Pathname
Prompt: Configuration file

Job Activity Trigger



Name	Expression Type	Expression
Job3_Warnings	Warning - (Conditional)	"Execution finished with warnings"
Ok	Custom - (Conditional)	Job_3.$JobStatus = DSJS.RUNOK Or Job_3.$JobStatus = DSJS.RUNWARN

Execute Command Stage

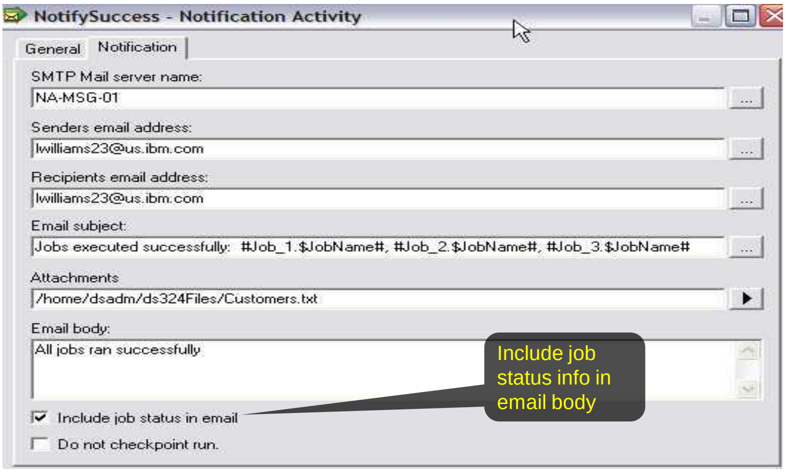


Command:
/home/dsadm/Echo_Script.sh

Parameters:
#Job_1.$JobName#=#Job_1.$JobStatus# #Job_2.$JobName#=#Job_2.$JobStatus# #Job_3.$JobName#=#Job_3.$JobStatus#

- Execute system commands, shell scripts, and other executables
- Use e.g. to drop or rename database tables

Notification Activity Stage



NotifySuccess - Notification Activity

General | Notification

SMTP Mail server name: NA-MSG-01

Senders email address: lwilliams23@us.ibm.com

Recipients email address: lwilliams23@us.ibm.com

Email subject: Jobs executed successfully: #Job_1.$JobName#, #Job_2.$JobName#, #Job_3.$JobName#

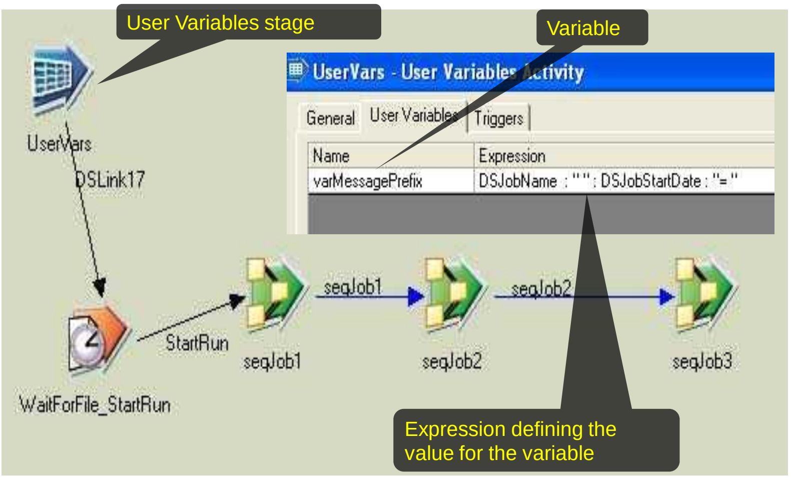
Attachments: /home/dsadm/ds324Files/Customers.txt

Email body: All jobs ran successfully

[✓] Include job status in email

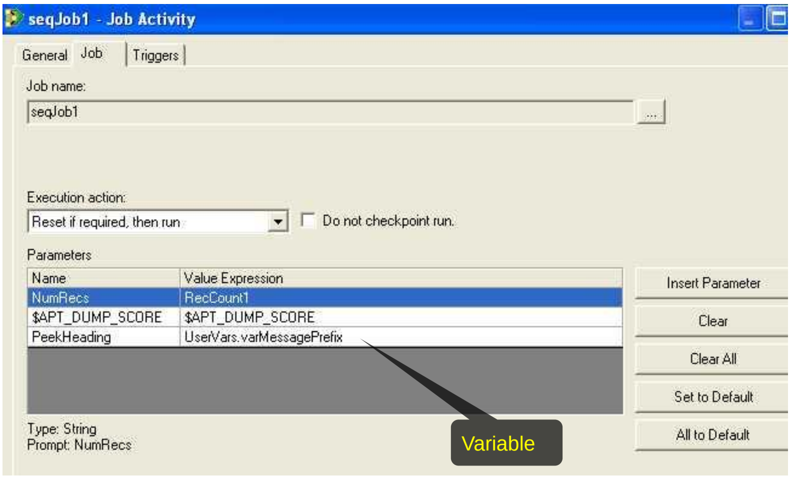
[ ] Do not checkpoint run.

User Variables Stage



Name	Expression
varMessagePrefix	DSJobName : " " : DSJobStartDate : "= "

Referencing the User Variable


seqJob1 - Job Activity

General | Job | Triggers

Job name: seqJob1

Execution action: Reset if required, then run

Do not checkpoint run.

Parameters

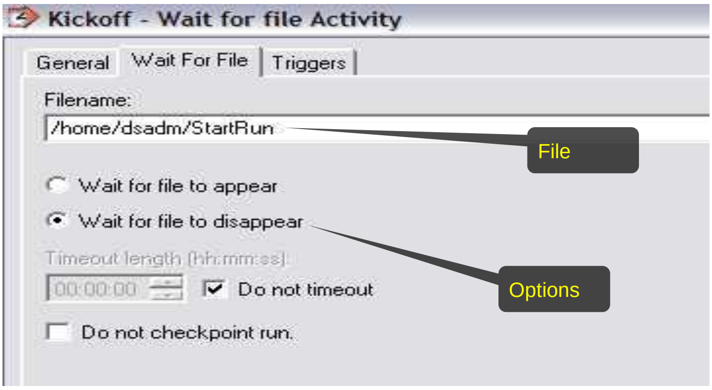
Name	Value Expression
NumRecs	RecCount1
$APT_DUMP_SCORE	$APT_DUMP_SCORE
PeekHeading	UserVars.varMessagePrefix

Type: String
Prompt: NumRecs

IBM InfoSphere Information Server

7.7 : Flow of Control Stages

Wait for File Stage



Kickoff - Wait for file Activity

General | Wait For File | Triggers

Filename: /home/dsadm/StartRun

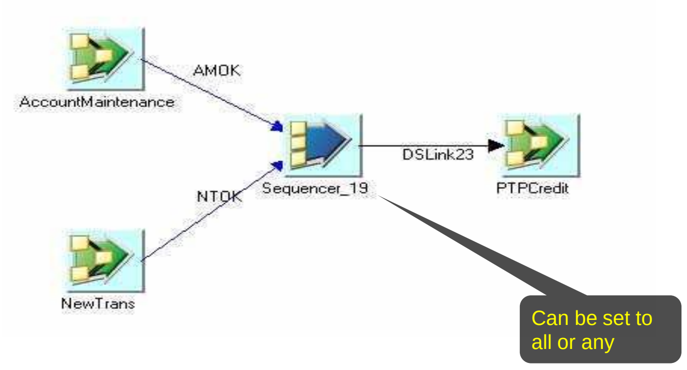
( ) Wait for file to appear
(●) Wait for file to disappear

Timeout length (hh:mm:ss): 00:00:00 [✓] Do not timeout

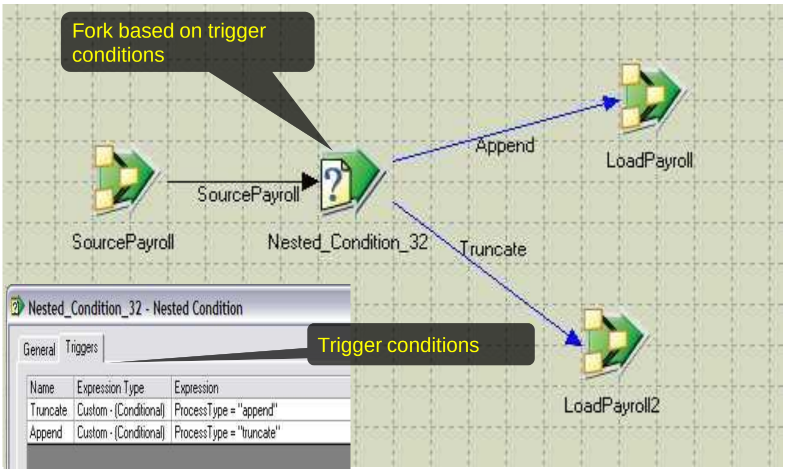
[ ] Do not checkpoint run.

Sequencer Stage

- Sequence multiple jobs using the Sequence stage

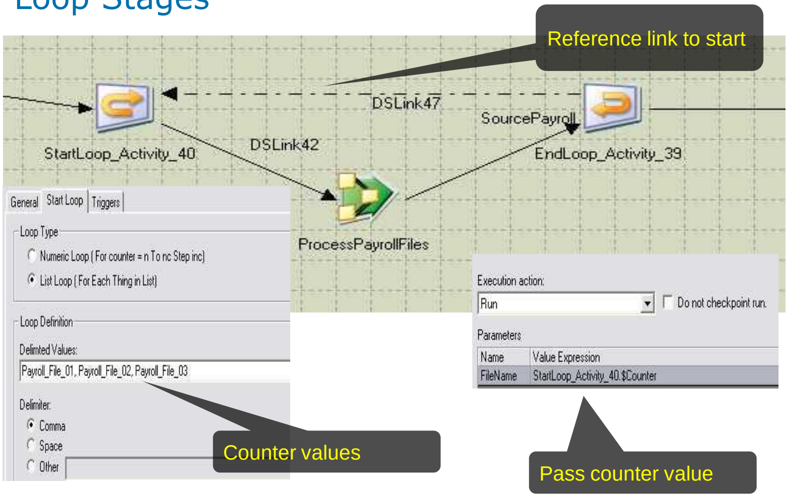


Nested Condition Stage



Name	Expression Type	Expression
Truncate	Custom - (Conditional)	ProcessType = "append"
Append	Custom - (Conditional)	ProcessType = "truncate"

Loop Stages



Loop Type
- Numeric Loop (For counter = n To nc Step inc)
- [✓ List Loop (For Each Thing in List)]

Loop Definition

Delimited Values: Payroll_File_01, Payroll_File_02, Payroll_File_03

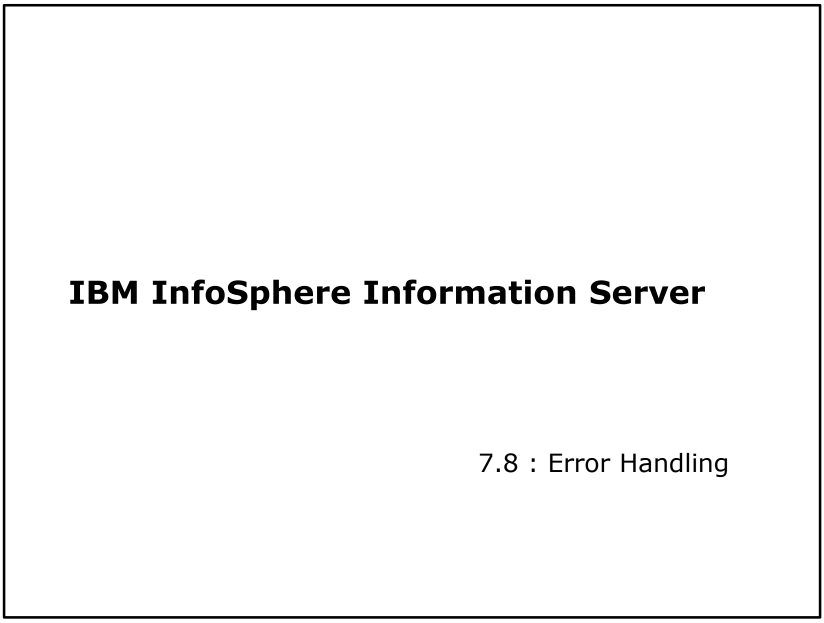
Delimiter: [✓ Comma] Space Other

Execution action: Run

Do not checkpoint run.

Parameters

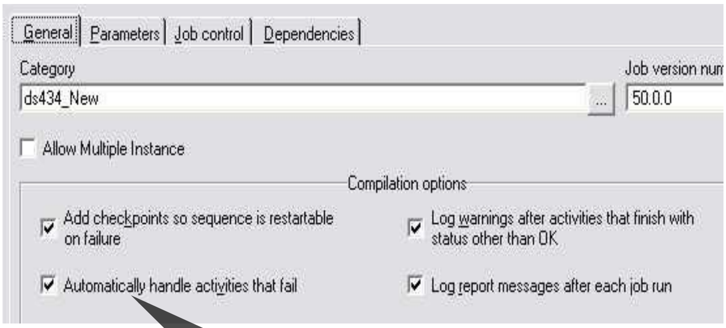
Name	Value Expression
FileName	StartLoop_Activity_40.$Counter



IBM InfoSphere Information Server

7.8 : Error Handling

Handling Activities that Fail



General | Parameters | Job control | Dependencies

Category: ds434_New

Job version num: 50.0.0

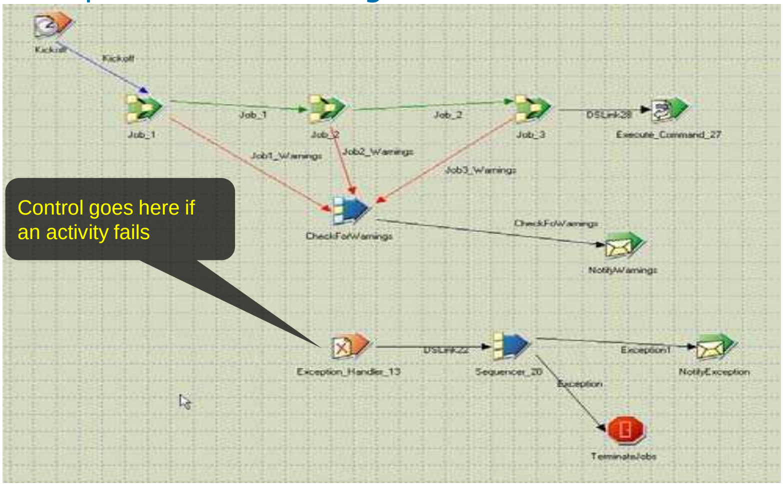
☐ Allow Multiple Instance

Compilation options

- ✓ Add checkpoints so sequence is restartable on failure
- ✓ Log warnings after activities that finish with status other than OK
- ✓ Automatically handle activities that fail
- ✓ Log report messages after each job run

Pass control to Exception stage when an activity fails

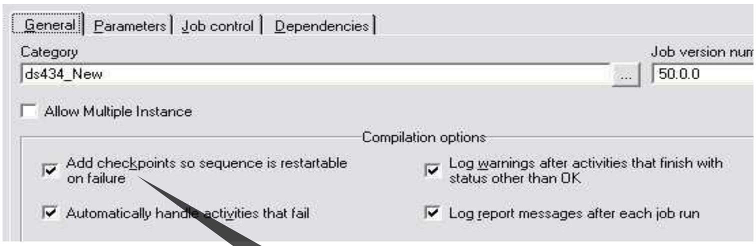
Exception Handler Stage



IBM InfoSphere Information Server

7.8 : Restart

Enable Restart



General | Parameters | Job control | Dependencies

Category: ds434_New

Job version num: 50.0.0

☐ Allow Multiple Instance

Compilation options

☑ Add checkpoints so sequence is restartable on failure

☑ Log warnings after activities that finish with status other than OK

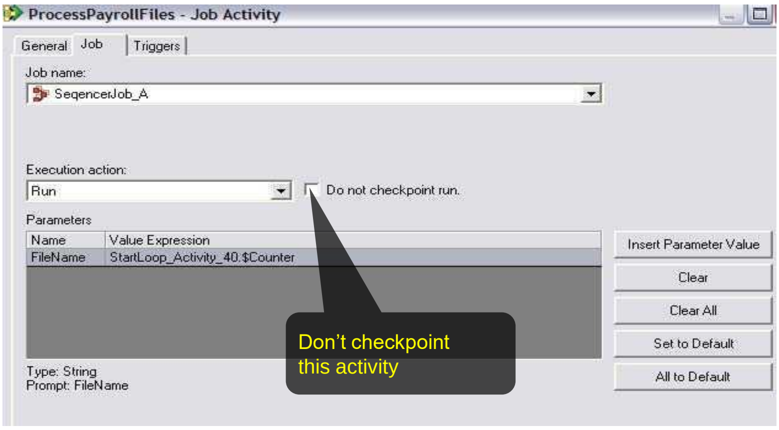
☑ Automatically handle activities that fail

☑ Log report messages after each job run

Enable checkpoints to be added

If a Sequence fails, and it is set to "Add check points so sequence is restartable on failure", then when the Sequence is re-run, activities that completed successfully in the prior run are skipped over (unless the "Do not checkpoint run" option was set for an activity).

Disable Checkpoint at a Stage



ProcessPayrollFiles - Job Activity

General | Job | Triggers

Job name: SeqencerJob_A

Execution action: Run

☐ Do not checkpoint run.

Parameters

Name	Value Expression
FileName	StartLoop_Activity_40.$Counter

Type: String
Prompt: FileName

Q & A

1. Which stage is used to run jobs in a job sequence?
2. Does the Exception Handler stage support an input link?

Q & A

- Job Activity stage
- No, control is automatically passed to the stage when an exception occurs.

Unit summary

- Having completed this unit, you should be able to:
- Use the DataStage Job Sequencer to build a job that controls a sequence of jobs
- Use Sequencer links and stages to control the sequence a set of jobs run in
- Use Sequencer triggers and stages to control the conditions under which jobs run
- Pass information in job parameters from the master controlling job to the controlled jobs
- Define user variables
- Enable restart
- Handle errors and exceptions