



DSXchange Webinar Series

February 20, 2009

DataStage Parameter Sets

Sponsored By:



STRATEGIES 4 DATA

Speaker Bio



Ray is a self-employed trainer and consultant for the IBM DataStage®, IBM UniVerse and IBM Red Brick Warehouse and DataStage® XE suites of products. Ray has taught advanced classes in the USA, the UK and Germany, and has been used frequently as a training consultant by IBM to conduct advanced in-house training classes. Additionally, Ray has presented training classes in almost every country in the Asia-Pacific region, and has been involved in technical presentations and implementations throughout the region.

Ray joined Prime Computer of Australia in 1986. He later joined VMARK Software (original developers of DataStage®) after Prime Computer sold its database businesses to VMARK. Ray's principal role with VMARK, and subsequently with Ardent, Informix and IBM was as a DataStage® trainer, but was also actively involved in technical support. While with VMARK and Ardent he was actively involved in the development of DataStage®, creating a complete training curriculum for use in the Asia-Pacific region. He has also developed training curriculum and train-the-trainer programs for the UniVerse® RDBMS, including its NLS (national language support) implementation. When Ardent sold its database businesses to Informix Software, Ray continued his involvement in Data Warehouse technology by becoming expert with the Red Brick Warehouse product; a database designed specifically for Data Warehouse (star schema) implementations.

When Informix was acquired by IBM Ray continued his concentration on training, while additionally focusing on Data Warehousing applications.

DSXchange Webinar Series

DataStage Parameter Sets

Synopsis:

Parameter Sets were introduced in version 8.0. A parameter set is a set of parameters stored as a single and separate object in the Repository. Rather than set up individual parameters in every job, you create a Parameter Set just once, and use that Parameter Set in every job that needs it. Because of the unified metadata handling in Information Server, Parameter Sets are available to other tools and products in the suite. For example a Data Connector that uses a Parameter Set and was created in DataStage might be used by the Information Analyzer product to provide access to data. Or vice versa, of course.

A Parameter Set is created in DataStage using the "New" tool or menu. Another way to create a Parameter Set in DataStage is to select one or more ordinary job parameters and click the Create Parameter Set command button. Nested Parameter Sets are not supported.

Each Parameter Set has a name and properties. General properties are short and long description. There is a collection of one or more parameter definitions (name, prompt text, parameter type, default value and help text). There is a collection of zero or more "values files" (name, value for each parameter in the set).

At run time the name of a values file may be supplied; if so the values stored in that file will be used in place of the parameters' default values.

To use a Parameter Set in a DataStage job simply click Add Parameter Set while in the Parameters page of the job properties dialog. It is also possible to view a selected Parameter Set while on this page - the Parameter Set editing dialog is opened in read-only mode for this purpose.

To refer to specific parameter in a Parameter Set use "dot notation", as `set_name.parameter_name`. For example a reference to a parameter called `RejectsDir` in a Parameter Set called `Directories` would be specified as `Directories.RejectsDir` (surrounded by `"#"` characters if used other than in an expression. Dot notation would also be used in API functions such as `DSGetParamInfo()` and in `-param` arguments to the `dsjob` command.

New DSXchange DataStage DVD's now available at the Learning Center.

QualityStage Rule Sets

Description:

Rule Sets provide the mechanism that underpins both Word Investigation and Standardization in QualityStage. In this presentation we show how to construct a QualityStage Rule Set. The various components of a Rule Set, namely the description file, the file dictionary, the classification table, the pattern-action language, and the various optional lookup tables – including rule set override tables – are reviewed and a particular rule set is examined to bring it all together. The tools within QualityStage for working with Rule Sets are also described.

Duration:

3 hours approximately

Audience:

This presentation is intended for QualityStage developers who have attended a training class such as IBM's QualityStage Essentials and who seek to know more about how to create custom Rule Sets .

Objectives:

- Having completed this module you will be able:
- to identify the components of a QualityStage Rule Set
- to state how Pattern Action Language defines parsing of tokens
- to test Rule Sets and to implement Rule Set Overrides

Contents:

- Introduction
- Parsing
- File Dictionary
- Classification
- Standardization
- Rule Set Overrides
- Building a Rule Set

QualityStage Pattern Action Language

Description:

Pattern Action Language is how one specifies in a QualityStage rule set how input records are to be processed after they have been parsed into separate tokens. This presentation covers both unconditional and conditional pattern elements, the various actions that can be associated with these, and reviews the structure of the pattern-action language file in a Rule Set.

Duration:

3 hours approximately

Audience:

This presentation is intended for QualityStage developers who have attended a training class such as IBM's QualityStage Essentials and who seek to know more about how to create pattern action language scripts in custom Rule Sets.

Objectives:

- Having completed this module you will be able:
- to construct unconditional patterns
- to construct conditional patterns
- to assemble a pattern action language file

Contents:

- Introduction
- Unconditional Patterns
- Conditional Patterns
- Actions
- Structure of PAT File

DataStage Job Sequences

Description:

Job sequences provide a mechanism for creating control hierarchies using the DataStage graphical user interface and paradigm. This presentation takes you through the various components available to use in DataStage job sequences, how to construct and compile them, how to detect and handle errors, automatic restart capability, and provides examples of how a number of tasks, such as initializing a key value, reading data from a file or processing a list of file names, are done.

Duration:

3 hours approximately

Audience:

This presentation is intended for DataStage developers who have attended a training class such as IBM's DataStage Essentials and who seek to know more about how to create job sequences to effect control of how a set of DataStage jobs, routines and operating system commands is executed.

Objectives:

- Having completed this module you will be able:
- to create and compile job sequences that fulfil a given requirement
- to construct a hierarchy of control of job sequences
- to implement error detection and handling in job sequences

Contents:

- Introduction
- Activities and Triggers
- Essential Activities
- "Programming" Activities
- Error Detection and Handling
- Examples

DataStage Change Detection

Description:

This presentation covers the various ways in which change detection can be performed within both parallel and server DataStage jobs. It begins by showing parallel job stage types that perform change detection then reviews the Slowly Changing Dimension stage that was introduced in version 8.0. It concludes by showing how change detection can be performed in server jobs.

Duration:

One hour approximately

Audience:

This presentation is intended for DataStage developers who have attended a training class such as IBM's DataStage Essentials and who seek to know more about how to select the most appropriate stage type for performing change detection tasks.

Objectives:

- Having completed this module you will be able:
- to differentiate between parallel job change detection stage types
- to implement change detection techniques in DataStage jobs

Contents:

- The Problem
- Change Detection Stage Types
- Change Capture and Difference
- Compare
- Slowly Changing Dimension
- Change detection in server jobs

Links

DSXchange Forum: www.dsxchange.com

DSXchange Member Forum: www.dsxchange.net

Contact Information:

