

Ascential DataStage™

WebSphere MQ Stage Guide

Version 2.0



This document, and the software described or referenced in it, are confidential and proprietary to Ascential Software Corporation ("Ascential"). They are provided under, and are subject to, the terms and conditions of a license agreement between Ascential and the licensee, and may not be transferred, disclosed, or otherwise provided to third parties, unless otherwise permitted by that agreement. No portion of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Ascential. The specifications and other information contained in this document for some purposes may not be complete, current, or correct, and are subject to change without notice. NO REPRESENTATION OR OTHER AFFIRMATION OF FACT CONTAINED IN THIS DOCUMENT, INCLUDING WITHOUT LIMITATION STATEMENTS REGARDING CAPACITY, PERFORMANCE, OR SUITABILITY FOR USE OF PRODUCTS OR SOFTWARE DESCRIBED HEREIN, SHALL BE DEEMED TO BE A WARRANTY BY ASCENTIAL FOR ANY PURPOSE OR GIVE RISE TO ANY LIABILITY OF ASCENTIAL WHATSOEVER. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL ASCENTIAL BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. If you are acquiring this software on behalf of the U.S. government, the Government shall have only "Restricted Rights" in the software and related documentation as defined in the Federal Acquisition Regulations (FARs) in Clause 52.227.19 (c) (2). If you are acquiring the software on behalf of the Department of Defense, the software shall be classified as "Commercial Computer Software" and the Government shall have only "Restricted Rights" as defined in Clause 252.227-7013 (c) (1) of DFARs.

© 2004, 2001-2003 Ascential Software Corporation. All rights reserved. DataStage®, EasyLogic®, EasyPath®, Enterprise Data Quality Management®, Iterations®, Matchware®, Mercator®, MetaBroker®, Application Integration, Simplified®, Ascential™, Ascential AuditStage™, Ascential DataStage™, Ascential ProfileStage™, Ascential QualityStage™, Ascential Enterprise Integration Suite™, Ascential Real-time Integration Services™, Ascential MetaStage™, and Ascential RTI™ are trademarks of Ascential Software Corporation or its affiliates and may be registered in the United States or other jurisdictions.

Adobe Acrobat is a trademark of Adobe Systems, Inc. IBM, MQSeries, and WebSphere are either registered trademarks or trademarks of IBM Corporation. Microsoft, Windows, Windows NT, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd. Other marks mentioned are the property of the owners of those marks. The software delivered to Licensee may contain third-party software code. See *Legal Notices* ([LegalNotices.pdf](#)) for more information.

How to Use This Guide

The WebSphere MQ stage lets Ascential DataStage use messaging software to read from and write to IBM WebSphere MQ message queues. Version 2.0 of the WebSphere MQ is compatible with Ascential DataStage Release 7.5.1.

Audience

This guide is intended for DataStage designers who create or modify jobs that use WebSphere MQ.

How This Book is Organized

The following table lists topics that may be of interest to you and it provides links to these topics.

To learn about	Read...
Functionality	"Functionality" on page 2
Terminology	"Terminology" on page 3
Installation	"Installing the Stage" on page 4
Using the WebSphere MQ stage	"Using the WebSphere MQ Stage" on page 4
Security	"Security" on page 11
Defining the WebSphere MQ connection	"Defining the WebSphere MQ Connection" on page 13
Defining input data	"Defining WebSphere MQ Input Data" on page 15
Defining output data	"Defining WebSphere MQ Output Data" on page 22
Using input and output column data elements	"Using Column Data Elements" on page 33

Related Documentation

To learn more about documentation from other Ascential products and third-party documentation as they relate to the WebSphere MQ stage, refer to the following sections/tables.

Ascential Software Documentation

Guide	Description
<i>Ascential DataStage Designer Guide</i>	General principles for designing jobs
<i>Ascential DataStage Server Job Developer's Guide</i>	Techniques for designing server jobs
<i>Ascential MetaStage User's Guide</i>	Information about Ascential MetaStage™
<i>Ascential DataStage NLS Guide</i>	Information about NLS and techniques for character-set mapping
<i>Ascential DataStage Plug-In Installation and Configuration Guide</i>	Information required to configure your system and install this stage

WebSphere MQ Documentation

Guide	Description
<i>IBM's WebSphere MQ Application Programming Guide</i>	Information about using IBM's WebSphere MQ

Conventions

Convention	Used for...
bold	Field names, button names, menu items, and keystrokes. Also used to indicate filenames, and window and dialog box names.
<code>user input</code>	Information that you need to enter as is.
<code>code</code>	Code examples

Convention	Used for...
<i>variable</i> or <variable>	Placeholders for information that you need to enter. Do not type the greater-/less-than brackets as part of the variable.
>	Indicators used to separate menu options, such as: Start >Programs >Ascential DataStage
[A]	Options in command syntax. Do not type the brackets as part of the option.
B...	Elements that can repeat.
A B	Indicator used to separate mutually-exclusive elements.
{ }	Indicator used to identify sets of choices.

Contacting Support

To reach Customer Care, please refer to the information below:

Call toll-free: 1-866-INFONOW (1-866-463-6669)

Email: support@ascentialsoftware.com

Ascential Developer Net: <http://developernet.ascential.com>

Please consult your support agreement for the location and availability of customer support personnel.

To find the location and telephone number of the nearest Ascential Software office outside of North America, please visit the Ascential Software Corporation website at <http://www.ascentialsoftware.com>.

Contents

Audience	iii
How This Book is Organized	iii
Related Documentation	iv
Ascential Software Documentation	iv
WebSphere MQ Documentation	iv
Conventions	iv
Contacting Support	v
Introduction	1
Functionality	2
Terminology	3
Configuration Requirements	4
Installing the Stage	4
Using the WebSphere MQ Stage	4
WebSphere MQ Stage Message Queues	5
Browsing the Queue	7
Processing Units of Work	7
Reply and Report Messages	11
Security	11
Defining the WebSphere MQ Connection	13
Connecting to a Queue Manager	14
Defining Character Set Mapping	14
Defining WebSphere MQ Input Data	15
About the Input Page	15
Specifying Transaction Control Information	20
Defining WebSphere MQ Output Data	22
About the Output Page	22
Using Column Data Elements	33

Introduction

Many enterprises are now turning to message-oriented, real-time transactional middleware for moving data between applications and other source and target configurations. The IBM WebSphere MQ family of products is messaging software that enables enterprises to integrate business processes.

The WebSphere MQ stage is a passive stage that offers a message-based solution to customers where messaging represents another form of source and target data. The WebSphere MQ stage lets Ascential DataStage read from and write to WebSphere MQ message queues. You can use this stage as:

- An intermediary between applications, transforming messages as they are sent between programs
- A conduit for the transmission of legacy data to a message queue
- A message queue reader for transmission to a non-messaging target

Message-based communication is powerful because two interacting programs do not have to be running concurrently as they do when operating in a classic conversational mode. You can use Ascential DataStage to transform and manipulate message contents, opened as rows and columns within the DataStage engine, like any other data stream. All of the benefits of using an ETL tool for data warehousing can now be applied to application integration.

The WebSphere MQ stage is the first step in providing real-time data warehousing and business intelligence. It supports multiple input and output links. Reference links are not supported because message data cannot be guaranteed to be persistent, and lookups are not key-based.

You can use the WebSphere MQ stage as a source or a target in any DataStage data flow diagram. It handles data in standard row and column format. As a message writer, the stage writes only datagram messages. As a message reader, the stage accepts all message types.

- If the reads are browse reads, the message remains on the queue.
- If transaction control for a unit of work applies, the message is removed from the queue but only after the commit following a successful write.

If a request message is read, you must ensure that another application reads and responds to the request message.

Functionality

The WebSphere MQ stage has the following functionality:

- Reads from and writes to WebSphere MQ message queues.
- Connects to a single queue manager, but can open several queues. You can associate each link with a different queue.
- Processes string-formatted messages.
- Provides read options to maintain the message on the queue, delete the message immediately after it is read, or delete the message when the job completes successfully.
- Provides termination conditions by processing a user-defined message type, a user-specified time-out period, or message count. (for message reading)
- Specifies the message descriptor fields that get sent on the input link. (for message writing)
- Controls how a message is to be delivered, such as its priority, persistence, and expiry data. (for message writing)
- Supports many-to-one and one-to-many row formats.
- Supports local units of work.
- Provides a custom GUI to facilitate the defining of meta data associated with the choice of message descriptor fields.
- Defines new data elements that specify transformations on message descriptor data.
- Supports NLS (National Language Support). See *Ascential DataStage NLS Guide*.
- Supports Ascential MetaStage™. See *Ascential MetaStage User's Guide*.

The following functionality is not supported:

- Reference links. These are unsupported since message data cannot be guaranteed to be persistent, and lookups are not key-based.
- Processing nonstring-based messages, such as triggers and other event-related messages.
- Global units of work.
- Meta data import.
- Data browsing, which is the ability to use the custom GUI to view sample native data.
- Request messages. All queue puts are datagram messages.

- Single-row, repeating-group row data.
- Complex message formats.
- MQ Series Client-only option.

Terminology

The following table lists the WebSphere MQ stage terms used in this document:

Term	Description
Datagram message	A message containing information for which no response is expected.
Message Queue	A synonym for queue.
Namelist	A WebSphere MQ object that contains a list of queue names.
Object authority manager (OAM)	The default authorization service for command and object management for WebSphere MQ on UNIX and Windows systems.
Queue	A WebSphere MQ object. Message queueing applications can write messages to and read messages from a queue.
Queue get, MQGET	An operation that retrieves a message from a queue. A read.
Queue Manager	A system program that provides queueing services to applications.
Queue put, MQPUT	An operation that writes a message to a queue. A write.
Request message	A message for which a reply is requested.
XML	A markup language for documents containing structured information. You can use it to publish these documents on the Web. XML is a popular format for messages in data integration applications.

Configuration Requirements

For configuration requirements supporting the installation, see *Ascential DataStage Plug-In Installation and Configuration Guide*.

Linux.

MQ libraries need to be linked in the following fashion in order to be used with DataStage. The MQ series 5.3 2.95.2 version libraries must be referenced at runtime. In `$DSHOME/dsenv`

```
EXPORT LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/mqm/lib2.95.2:/usr/lib
```

Also, the MQ series 2.95.2 library links must be created as

```
ln -s /opt/mqm/lib/2.95.2/libimqb23gl.so /opt/mqm/lib/libimqb23gl.so
ln -s /opt/mqm/lib/2.195.2/libimqb23gl_r.so /opt/mqm/lib/libimqb23gl_r.so
ln -s /opt/mqm/lib/2.95.2/libimqs23gl.so /opt/mqm/lib/libimqs23gl.so
ln -s /opt/mqm/lib/2.95.2/libimqs23gl-r.so /opt/mqm/lib/libimqb23gl-r.so
```

Finally, the libstdc++ link also must be created as

```
ln -s /usr/lib/libstdc++-2-libc6.1-1-2.9.0.so \
/usr/lib/libstdc++-libc6.1-2.so.3
```

Installing the Stage

For instructions and information supporting the installation, see *Ascential DataStage Plug-In Installation and Configuration Guide*.

Note The MQ Series Client-only option is unsupported.

Using the WebSphere MQ Stage

The primary purpose of the WebSphere MQ stage is to read from or write to WebSphere MQ message queues. See "[WebSphere MQ Stage Message Queues](#)" on page 5 for an explanation of message queues.

To read data from a WebSphere MQ queue, you can use one of two approaches:

- Browse the queue (see "[Browsing the Queue](#)" on page 7), leaving the message intact or destroying the message
- Use transaction control under unit of work to read a message from a queue and write it successfully before a commit destroys the original message (see "[Processing Units of Work](#)" on page 7)

WebSphere MQ Stage Message Queues

The following sections describe queue managers, queue and message details, and rows in messages as well as formats and security.

Queue Manager

The queue manager controls one or more queues. When you design a job, you specify appropriate stage properties to connect to the queue manager. Each stage instance can connect to only a single queue manager.

Since you can open any number of queues, however, the queue name is a link property. This required property lets a stage instance open a single queue per link for reading or writing. You can use the Ascential DataStage Director to validate the connection by using the supplied values for the queue and queue manager names.

Namelists are supported for input links. A namelist is a WebSphere MQ object that contains a list of queue names. If you specify a namelist instead of a queue name, the stage writes messages to each queue defined in the namelist by dynamically creating a distribution list from these queues. Distribution lists let you use a single write to send a message to multiple destinations. You can use a single open to open multiple queues and use a single write to send the message to each of these queues.

For more information about the input and output links, "[Defining WebSphere MQ Input Data](#)" on page 15 and "[Defining WebSphere MQ Output Data](#)" on page 22.

Queue and Message Details

You can specify link properties to define additional queue and message details. The column meta data describes how to parse the message content into one or more columns. Since complex message formats are not supported, the WebSphere MQ stage parses all messages as fixed-length records. The meta data defines the length of each field. The WebSphere MQ stage supports messages that are equal to or shorter than the field lengths defined in the meta data. For more information about column widths and data lengths, see [page 32](#).

Because there is currently no standard for describing and retrieving the structure and arrangement of a WebSphere MQ message, you must handle this in your application design. The WebSphere MQ stage cannot obtain the structure of the message dynamically.

Actual column definitions are retrieved directly from the application programs, their supported design tools or entered manually. For messages using XML content, the DataStage support for Document Type Definitions (DTD) and XML document meta data defines message details.

Rows in a Message

In simple terms, Ascential DataStage reads and processes a row for every queue get operation or executes a queue put for every row that the stage receives from another part of the job.

However, for some applications, there is no one-to-one correlation between a relational row and a single transaction contained within a message. Furthermore, the WebSphere MQ message can have its own hierarchical structure. There can be one physical row per message, but the row itself can contain multiple repeating groups.

WebSphere MQ messages can be very large. A single message can be an entire answer set or the contents of a file. You define the length of a row when you design the link of the DataStage job in one of the following ways:

- **Number of columns.** Multiple rows can be enclosed in a single message. The total number of columns, including their byte lengths, equals the length of the row. The stage releases rows to the DataStage engine based on this length as it reads messages. See "[Columns Tab](#)" on page 30 for more information about the columns.
- **CRLF.** If CRLF is available in ASCII messages, it optionally indicates the end of a row when reading messages. This means that for one queue get, there can be many rows released down a link for processing. The **Ignore end-of-record** box on the **Output** page specifies whether to treat CRLF as a row terminator.
- **Rows per Message.** When writing WebSphere MQ messages, the **Rows per Message** box on the **Input** page specifies when to execute a queue put.

For more information about the **Input** and **Output** pages, see "[About the Input Page](#)" on page 15 and "[About the Output Page](#)" on page 22.

XML Format

XML is a popular format for messages in data integration applications. The hierarchical nature of XML implies the possibility that a single queue get equals a single row.

It also implies that the row has a complex internal structure that includes repeating groups. This may be true for other message

formats as well. In these cases, the WebSphere MQ stage simply processes these messages as a single row and lets the row be transformed by existing technologies within Ascential DataStage, such as the XML reader stage. The WebSphere MQ stage does not support complex message parsing.

Browsing the Queue

One method of reading messages from a queue is to browse the queue. With a browse mode read, you choose whether to remove the message from the queue. A nondestructive read leaves the message intact. A destructive read destroys the message under specific circumstances.

After the WebSphere MQ stage reads the message, it releases the row or message on its output link to:

- Another WebSphere MQ stage
- A Transformer stage
- Any other active or passive stage

The row or message can then be processed as appropriate for the application.

Processing Units of Work

An alternative method of handling queues includes not only reading a message from a queue but removing that message from the queue after the message is successfully processed. To do this, the WebSphere MQ stage incorporates transaction control into queue management using units of work. The WebSphere MQ stage supports the local definition of a unit of work that is exclusive to and within the context of the connection to a single queue manager. This is useful when Ascential DataStage reads from a queue, translates message contents, then delivers the information to another queue. Transaction control requires the use of a transformer stage, and transaction control must be turned on (see [Enable transaction grouping on page 20](#)).

Note Global units of work cross MQ and RDBMS boundaries, such as reading a message, then performing an SQL insert. These global units of work are not currently supported. Only local units of work are currently supported.

Ascential DataStage processes a unit of work as follows:

- 1 Reads the queue without destruction as a queue-browse operation.

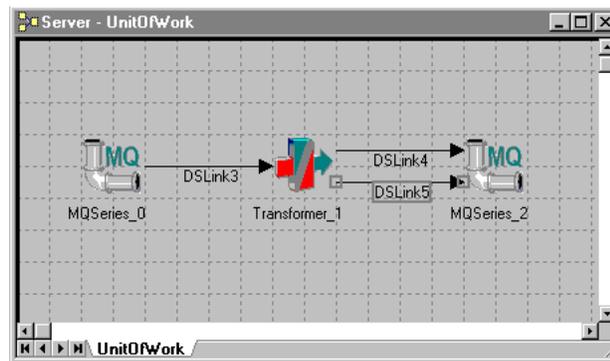
Messages remain on the source queue after the read. This is done because it is not prudent to remove a message from the queue until the message has been processed successfully. The reader WebSphere MQ stage has no way of determining if processing is successful.

- 2 Processes the message.
- 3 Executes a second read of the message.
- 4 Writes the message to the target queue.
- 5 Executes a queue commit.

If the job fails before the commit, the original message is still available in the source queue. This functionality is supported by the syncpoint control options for queue get and queue put and the commit and rollback calls. If the job is successful, the original message is removed as part of the commit.

Job Scenario

The following job shows how the WebSphere MQ stage handles a local unit of work. For more information about input and output links, see "[General tab](#)" on page 16 for the **Input** page and "[General Tab](#)" on page 22 for the **Output** page.



A Transformer stage splits the physical message into logical rows, which can be sent to a WebSphere MQ writer stage on separate input links. You can coordinate these message writes in a local unit of work because the input link rereads the original message.

The **Transaction Handling** tab on the **Input** page defines the role of each link in the transaction control group. For more information about the **Transaction Handling** tab on the **Input** page, see [page 19](#).

The first link, DSLink4, defines the controlling link, which carries the message identifier. Unconventionally, it does a read of the message identified by the message identifier rather than a write.

A successful write on DSLink5 commits the unit work, removing it from the source queue. A failure rolls it back, which causes the message to remain on the source queue.

If more than two queues receive the original message or a transformation of the message, it implies that there are more than two input links to the WebSphere MQ stage. In this case, only the last link causes a commit on success. A failure on any input link, including the controlling link, causes a rollback, leaving the original message intact.

These next sections give detailed information about how the WebSphere MQ stage incorporates unit of work and transaction control while doing the following:

- Reading queues in browse mode
- Writing messages to destination queues using message identifiers
- Executing a queue commit for a local unit of work

Reading Queues

The WebSphere MQ stage reads messages from a queue in browse mode and releases the row or message on its output link to a Transformer stage. A browse mode read is nondestructive.

The row contains a unique message identifier that moves along the data flow to the Transformer stage that coordinates transaction control. This Transformer stage branches into two or more output links that enter the same WebSphere MQ stage for writing.

Writing Messages to Target Queues

The Transformer stage defines the set of output links as a transactional group. One of the links in the transactional group is defined as the controlling link. The only column that must be defined on the controlling link is the message identifier.

Executing a Queue Commit

In a WebSphere MQ stage with multiple input links, the controlling link uses the unique message identifier provided on the link to reread the original message in the source queue.

The queue read is then executed under syncpoint control, which implies the start of a transaction. The stage writes the message to its target queues for each of its subsequent links in the transaction group.

If any queue writes fail, the WebSphere MQ stage executes a queue rollback based on the reread message, and the original message remains on the queue. However, if all the queue writes succeed, the stage executes a queue commit, and the original message is removed from the queue.

Transactional control is unsupported when there is a one-to-many relationship between a message and Ascential DataStage rows. Avoid this situation by doing the following:

- Use the **Ignore end of record** field on the **General** tab of the **Output** page to ignore CRLF as a logical row terminator (see "[General Tab](#)" on page 22)
- Set the total combined length of the columns designated by **Number of columns** to be at least the size of the message

Troubleshooting

This section describes possible problems you may encounter and workarounds to resolve them.

Infinite loops

If you read from and write to the same queue in the same process, the messages are continuously written into the queue until the maximum number as specified by **Message Limit** is reached. An infinite loop can result. To resolve this problem, do one of the following:

- **Message Limit.** Specify a positive value to cause the stage to stop reading messages. For more information about the parameters on the **General** tab of the **Output** page, see "[General Tab](#)" on page 22.
- **Staging area.** Use a Sequential File stage as a staging area for temporary storage. Then add another process to write messages back to the queue.

Queue Manager configuration

Only one connection to one queue manager is allowed for one process. To resolve this limitation, do one of the following:

- **Remote queue definition.** Configure a queue belonging to a second queue manager as a remote queue definition for the primary queue manager.
- **Cluster.** Configure the primary and remote queue managers as a cluster. This lets a single local queue manager access queues belonging to different queue managers.

- **Staging area.** Use a Sequential File stage as a staging area for temporary storage. Then add another process to write messages back to the queue. This way, each process has its own connection to its own queue manager.

Reply and Report Messages

When a queue read occurs, some messages read by the stage can request a reply or report message in return. Since these requests are usually specific to the application, the WebSphere MQ stage does not return any reply or report messages requested by the sending application.

However, in the job scenario described on [page 8](#), messages within a job retain the message descriptor information that is defined by the message originator. When the message reaches its intended destination, the destination application services the replies and reports requested by the originating application.

Security

The object authority manager (OAM) is the default authorization service for command and object management for WebSphere MQ on UNIX and Windows systems. It authorizes access to queue managers and queues using access control groups, which correspond to user groups for operating systems. The user identity of the application determines the access to WebSphere MQ objects. You can replace the OAM or run it with your own security service.

When a message is put on a queue, the queue manager supplies a user name in the message descriptor. The default OAM then authorizes access based on this user name. The queue manager gets the user name from the operating system by default. However, the application can supply its own user id, which must be a valid operating system user.

You can use the stage properties **User Name** and **Password** to change the identity of the user running the job. Once authenticated, the stage uses the user name to connect to the queue manager as follows:

- **For message writes.** The identity context of the message is modified to reflect the specified user.
- **For message reads.** The rights associated with the specified user are used by the queue manager to determine message availability.

Windows. For Windows, the Ascential DataStage user that executes a job must have the 'create a token object' policy granted by a Windows administrator. Without this user policy, the user cannot impersonate the user specified in the user name property of the WebSphere MQ stage.

Administration groups. By default, users belonging to an administration group have complete access to a queue manager and local queues for a system. Examples of an administration group are:

- Administrator on Windows
- *root* on UNIX
- The WebSphere MQ mqm group on both operating systems

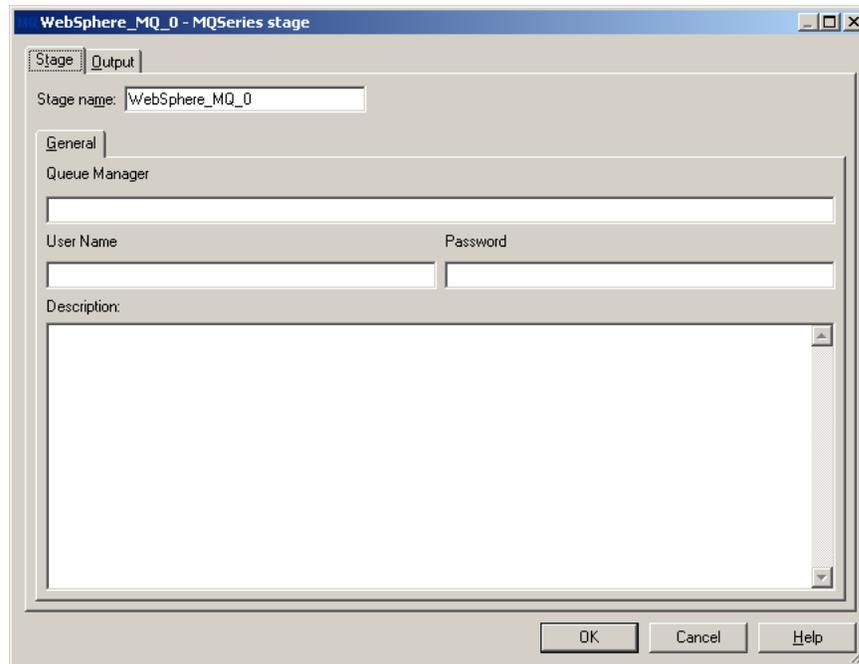
Other users must be given explicit access to the various WebSphere MQ objects.

setmqaut utility. If you are using the OAM, use the setmqaut utility to grant the required permissions to other operating system users. The following minimal permissions are required for input and output links. These permissions are required for a nonadministrative user to run jobs that contain a WebSphere MQ stage instance:

- Output links (read)
 - Connect on the queue manager
 - Get and browse on the queue
- Input links (write)
 - Connect on the queue manager
 - Put and passall on the queue
- Input links with a namelist (distribution list write)
 - Connect on the queue manager
 - Inq on the namelist
 - Put and passall on all queues names in the namelist
- Input links within local units of work (for details on units of work, see "[Processing Units of Work](#)" on page 7)
 - See the previous permissions, depending on the destination (queue or namelist)
 - Get on the queue for the syncpoint read

Defining the WebSphere MQ Connection

When you use the stage GUI to edit a WebSphere MQ stage, the **MQSeries Stage** dialog box appears.



This dialog box has the **Stage**, **Input**, and **Output** pages (depending on whether there are inputs to and outputs from the stage). In this case, there are no outputs from the stage.

- **Stage.** This page displays the name of the stage you are editing. The **General** tab defines the WebSphere MQ connection. For details, see ["Connecting to a Queue Manager"](#) on page 14.
The **NLS** tab defines a character set map to use with the stage. This tab appears only if you have installed NLS for Ascential DataStage. For details, see ["Defining Character Set Mapping"](#) on page 14.
- **Input.** This page is displayed only if you have an input link to this stage. It specifies the queue or namelist to which messages are written. It also specifies priorities, types of messages, the persistence of messages, and how messages are written to a queue.
- **Output.** This page is displayed only if you have an output link to this stage. It specifies the conditions to be met for the stage to stop reading messages. It also specifies how to handle the end of logical DataStage rows and message tracking options.

The main phases in defining a WebSphere MQ stage from the **MQSeries Stage** dialog box are as follows:

- 1 Connect to a queue manager (see the next section).
- 2 Optional. Define a character set map (see [page 14](#)).
- 3 Define the data on the input links if a target stage (see [page 15](#)). Or define the data on the output links if a source stage (see [page 22](#)).

Click **OK** to close this dialog box. Changes are saved when you save the job design.

Connecting to a Queue Manager

The WebSphere MQ connection parameters are set on the **General** tab on the **Stage** page. To connect to a WebSphere MQ queue manager, specify the following information:

- **Queue Manager.** The name of the queue manager to which the stage connects. This is a required field. For more information, see "[Queue Manager](#)" on [page 5](#).
- **User Name.** The alternate user name you can use to connect to the queue manager. If this field is left empty, the DataStage connection information is used. **User Name** is not active on UNIX platforms.
- **Password.** A password for the specified user name. It is ignored if **User Name** is omitted. **Password** is not active on UNIX platforms.
- **Description.** Optional. A description of the purpose of the stage.

Defining Character Set Mapping

You can define a character set map for a Plug-in stage. Do this from the **NLS** tab that appears on the **Stage** page. The **NLS** tab appears only if you have installed NLS.

Specify information using the following button and fields:

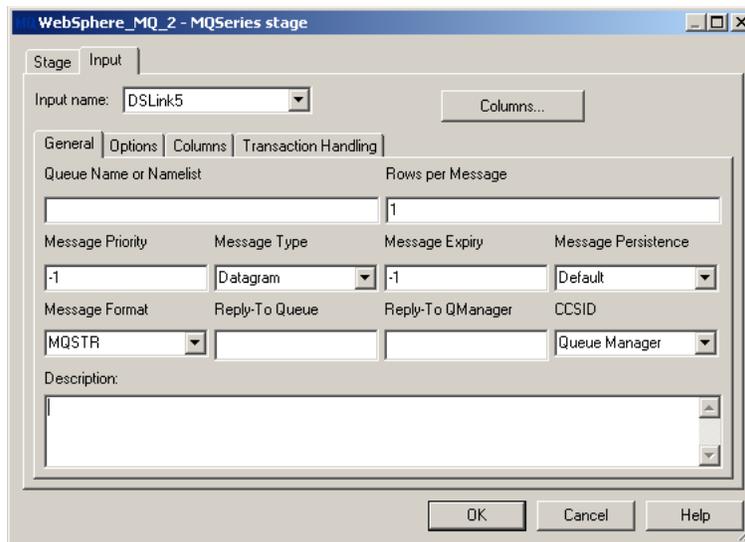
- **Map name to use with stage.** Defines the default character set map for the project or the job. You can change the map by selecting a map name from the list.
- **Use Job Parameter...** . Specifies parameter values for the job. Use the format *#Param#*, where *Param* is the name of the job parameter. The string *#Param#* is replaced by the job parameter when the job is run.
- **Show all maps.** Lists all the maps that are shipped with Ascential DataStage.

- **Loaded maps only.** Lists only the maps that are currently loaded.

For more information about NLS or job parameters, see *Ascential DataStage Designer's Guide* or *Ascential DataStage NLS Guide*.

Defining WebSphere MQ Input Data

When you write messages to a message queue, the WebSphere MQ stage has an input link. Define the properties of this link and the column definitions of the data on the **Input** page in the **MQSeries Stage** dialog box of the stage GUI.



About the Input Page

The **Input** page has an **Input name** field, the **General**, **Options**, **Columns**, and **Transaction Handling** tabs, and the **Columns...** button:

- **Input name.** The name of the input link. Choose the link you want to edit from the **Input name** list box. More than one input name exists when more than two queues receive the original message or a transformation of it. The first link can define the controlling link, which does the queue read (the reread of the original message). But the order of link execution can be modified in the transformer. To determine the order of execution:
 - Pass the cursor on a link in the job flow diagram. A ToolTip shows the link's order of execution.
 - Open the Transformer Editor. The sequence of the output links in the right pane indicates the order of execution.

To change the execution sequence, click **Output Link Execution Order**. The **Transformer Stage Properties** dialog box appears open to the **Link Ordering** tab on the **Stage** page.

- Select the **Transaction Handling** tab on the **Input** page. Select **Enable Transaction Grouping**. The links are listed in the order of execution.

- **Columns...** . Click the button to display a brief list of the columns designated on the input link. As you enter detailed meta data in the **Columns** tab, you can leave this list displayed.

General tab

This tab is displayed by default. It contains the following fields:

- **Queue Name or Namelist.** The name of a queue or namelist to which messages are written. If you specify a namelist, messages are written to each queue in the namelist. If in transactional mode, the message writes are not committed unless the writes to all the queues in the namelist are successful. See "[WebSphere MQ Stage Message Queues](#)" on page 5 for more information about namelists.
- **Rows per Message.** The number of rows the stage buffers before executing a write. The default is one write per row arriving on the input link.
- **Message Priority.** The priority of the message that is written to the queue. A value of -1 causes the message to be written with the default priority of the queue manager. Zero is the lowest priority. If the specified priority exceeds the maximum priority supported by the queue manager, the queue manager accepts the message, placing it on the queue at the maximum priority for the queue manager. The queue manager returns a warning in this case. For more information about the queue manager, see "[Queue Manager](#)" on page 5.
- **Message Type.** The type of message to be written. The list box displays the WebSphere MQ system-defined message types: **Request**, **Reply**, **Report**, and **Datagram**. You can also enter a numeric value representing a user-defined message type. The default is a datagram message.
- **Message Expiry.** The length of time in tenths of a second that a message remains on the queue. If the message remains on the queue longer, it is discarded. A value of -1 represents an indefinite amount of time. A value of 0 is not allowed.

- **Message Persistence.** The persistence of messages written to the queue. Choose one of the following values:
 - **Default.** A message is placed on the queue with the default persistence of the queue manager.
 - **Persistent.** A message survives restarts of the queue manager.
 - **Non-persistent.** A message does not survive restarts of the queue manager.
- **Message Format.** The format of messages written to the queue. Choose one of the following values:
 - **MQSTR**
 - **MQHRF2**

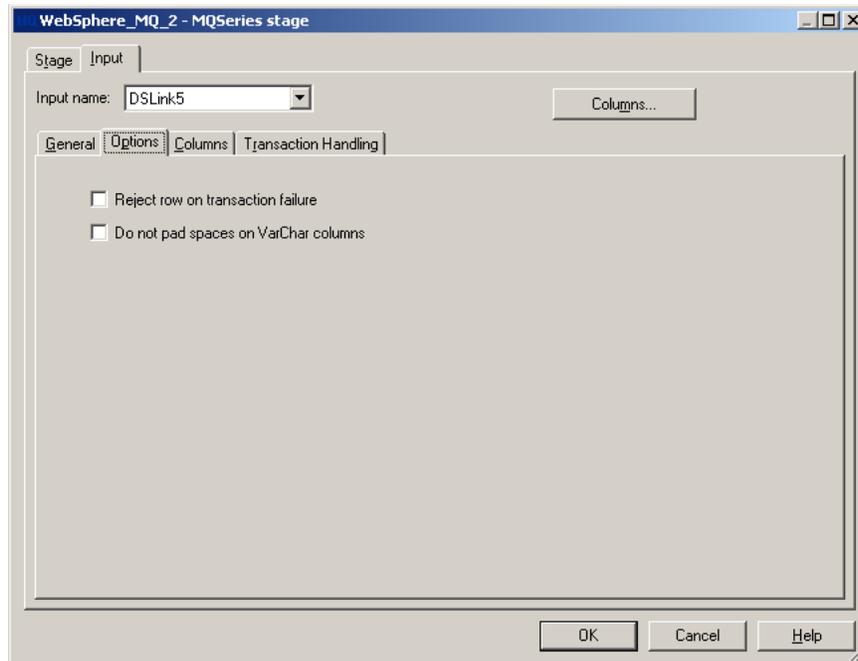
The default is **MQSTR**.

- **Reply-To Queue.** The name of the queue. If **Message Type** is **Request**, you have the option to provide the name of a queue.
- **Reply-To QManager.** The name of the queue manager. If **Message Type** is **Request**, you have the option to provide the name of a queue manager.
- **CCSID.** The character set identifier of the character data in the message. Choose one of the four default values described below or provide a value of your choice.
 - **Queue Manager.** The character set identifier of the queue manager.
 - **Default.** The default coded character set identifier.
 - **Inherit.** The inherited character-set identifier of this structure.
 - **Embedded.** The embedded character set identifier.

The default is **Queue Manager**. See IBM WebSphere MQ documentation for additional information.

Options Tab

This tab contains options controlling how a message is to be written.



It contains the following fields:

- **Reject row on transaction failure.** The handling of a message when an attempt to write the message fails. If selected and a put message fails, a job continues to run except in the following cases:
 - MQRC_CONNECTION_BROKEN
 - MQRC_MISSING_REPLY_TO_Q
 - MQRC_NOT_OPEN_FOR_INPUT
 - MQRC_PUT_INHIBITED
 - MQRC_Q_DELETED
 - MQRC_Q_FULL
 - MQRC_Q_MGR_NOT_AVAILABLE
 - MQRC_Q_MGR_STOPPING
 - MQRC_CONNECTION_STOPPING
- **Do not pad spaces on VarChar columns.** The treatment of a message when its length is less than the maximum length of the VarChar column. If selected, the column is not padded with spaces at the end of the message.

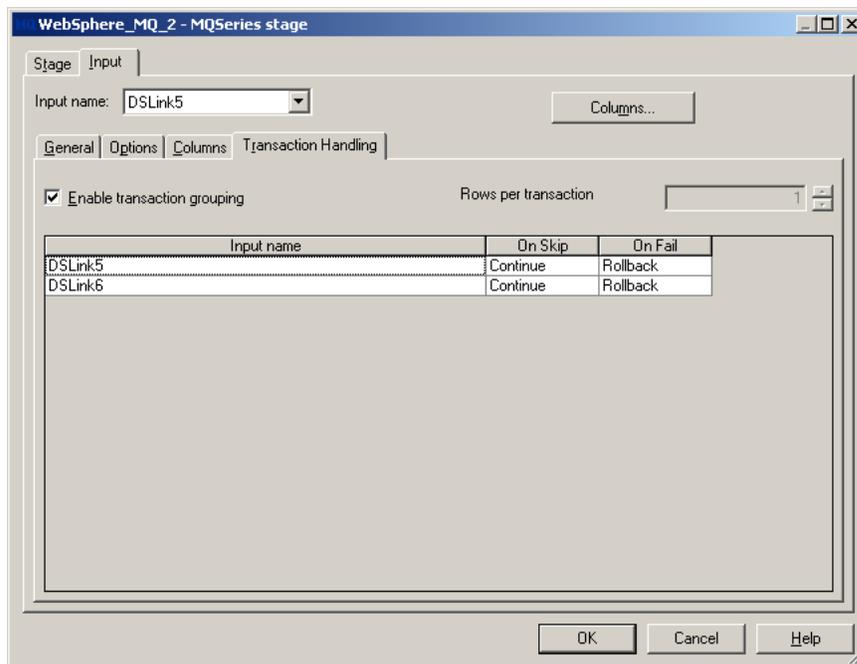
Columns Tab

This tab contains the column definitions for the data written to the message queues. The **Columns** tab behaves the same way as the **Columns** tab in the ODBC stage. For a description of how to enter and edit column definitions, see *Ascential DataStage Designer Guide*.

Note Rules for column widths are documented on [page 32](#).

Transaction Handling Tab

This tab contains the parameters that define the role of each link in the transaction control group, letting you view the transaction handling features of the stage as it writes to the data source.



Using transaction handling depends on whether **Enable transaction grouping** is selected. You can specify the number of rows written before each commit.

For information about...	See...
The job scenario for DSLInk4 and DSLInk5 as shown below	"Processing Units of Work" on page 7
Specifying transaction control information	"Specifying Transaction Control Information" on page 20

The **Transaction Handling** tab contains the following fields:

- **Enable transaction grouping.** A check box, which if selected, displays the grid with details of the transaction group to which the currently selected input link belongs. **Enable transaction grouping** is available if you have at least two input links. The check box is cleared by default.
- **Rows per transaction.** The number of rows written before the data is committed to the data table. Set the value explicitly to 1 to avoid causing the local units of work configuration to fail.
- **On Skip.** The action taken after a successful write. It specifies whether to continue or to roll back if a link is skipped because of an unsatisfied constraint in the preceding Transformer stage. Choose Continue or Rollback from the list. **On Skip** is available if **Enable transaction grouping** is selected.
- **On Fail.** The action taken after a failed write. It specifies that a transaction is rolled back. A rollback causes the message to remain on the source queue. **On Fail** is available if **Enable transaction grouping** is selected.

Specifying Transaction Control Information

As previously discussed, you can associate multiple input links writing to a single data source together as a transaction group. The transaction grouping feature is turned on and off using the **Enable transaction grouping** check box on the **Transaction Handling** tab (it is cleared by default).

If you clear **Enable transaction grouping**, you can enter a suitable value in the **Rows per transaction** field on the **Transaction Handling** tab. This is the number of rows written before the data is committed to the data table. The default value is 0, that is, all the rows are written before being committed to the data table.

If transaction grouping is enabled, the following rules govern the grouping of links:

- All the input links in the transaction group must originate from the same Transformer stage.
- The ordering of the links within the transaction group is determined in the preceding Transformer stage.
- A transaction group cannot use a **Rows per transaction** value other than 1.

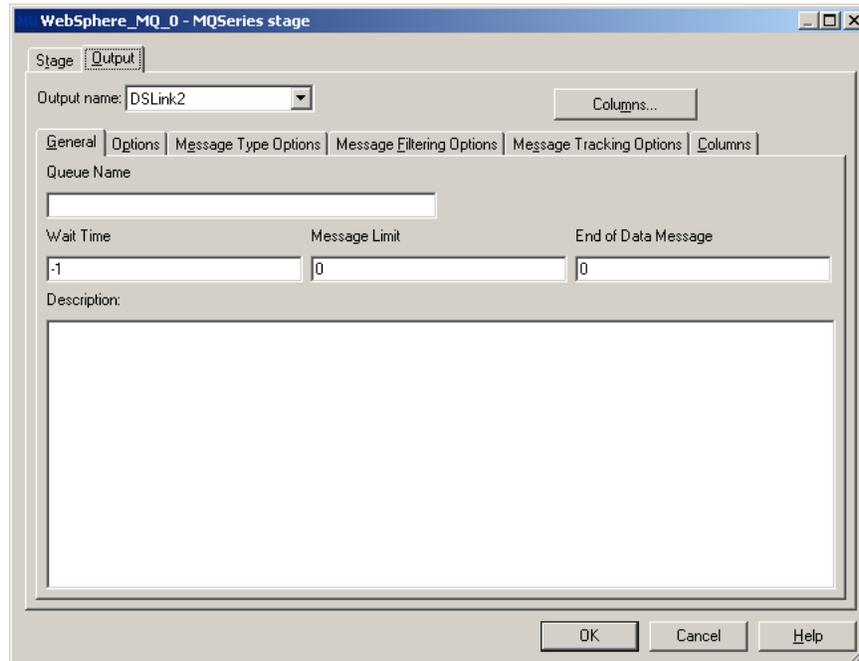
Note the following facts about transaction groups:

- A transaction starts at the beginning of each iteration of the Transformer stage preceding the WebSphere MQ stage. Any uncommitted changes left over from a previous transaction are rolled back.
- The links in the transaction group are processed in the order specified by the Transformer stage. Individual links are skipped depending on the constraints specified in the preceding Transformer stage.
- Each link in the transaction group can specify whether to rollback on failure. A rollback on any link causes the transaction to be abandoned, and any subsequent links in the group are skipped.
- Each link in the transaction group can be set to rollback if a constraint on that link is not met. Again, such a rollback causes the transaction to be abandoned, and any subsequent links in the group are skipped.
- The row counter for each link is incremented only if the message write associated with the link executes successfully and the transaction is successfully committed.
- The transaction ends after the last link in the transaction group is processed unless a preceding link performs a rollback. In this case, the transaction ends there.

For an example using links in a transaction group, see "[Processing Units of Work](#)" on page 7.

Defining WebSphere MQ Output Data

When you *read* data from a WebSphere MQ queue, the WebSphere MQ stage has an output link. The properties of this link and the column definitions are defined on the **Output** page in the **MQSeries Stage** dialog box.



About the Output Page

The **Output** page has an **Output name** field, the **General**, **Options**, **Message Type Options**, **Message Filtering Options**, **Message Tracking Options**, and **Columns** tabs, and the **Columns...** button.

- **Output name.** The name of the output link. Choose the link you want to edit from the **Output name** list box. This list box displays all the output links.
- **Columns...** . Click the button to display a brief list of the columns designated on the output link. As you enter detailed meta data in the **Columns** tab, you can leave this list displayed.

General Tab

This tab is displayed by default. It provides the interface for entering queue names, the job controls necessary to stop reading messages, and options to track messages.

A WebSphere MQ queue stage must know when to stop reading messages from the queue. Since messages can be delivered

predictably or randomly, it is difficult to identify the logical end of the data set.

Output link properties control when the stage stops reading messages from the queue. These properties are not mutually exclusive. The stage returns an end of data message when the first of these conditions is met. You can define the following:

- The time to wait for a message to arrive
- The message sent when the end of the data is reached
- The number of messages read from the queue

Output link properties also provide read options to

- Maintain messages on the queue
- Delete each message immediately after it is read
- Delete messages when the job completes successfully

The **General** tab contains the following fields:

- **Queue Name.** The name of the queue from which messages are read.
- **Wait Time.** The number of seconds to wait for a message to be read from the queue. If this period elapses and no messages are available, the job finishes. Choose one of the following values:
 - 0 specifies no wait
 - -1 specifies an indefinite wait period (default)
 - Positive integers specify to wait *n* seconds

If you specify a wait time of more than 5 minutes, the time is incremented by intervals of 5 minutes, for example, 7 minutes is incremented to a wait time of 10 minutes.

- **Message Limit.** The number of messages to be read from the queue. A positive integer (a value greater than zero) for this field causes the stage to stop reading messages after the specified number has been read, even if messages remain on the queue. The default value 0 indicates that **Message Limit** does not indicate the end of the data.

Note A warning occurs that the job has no termination conditions if all the following conditions are met:

- Wait Time is -1
- Message Limit is ≤ 0
- End of Data Message is 0

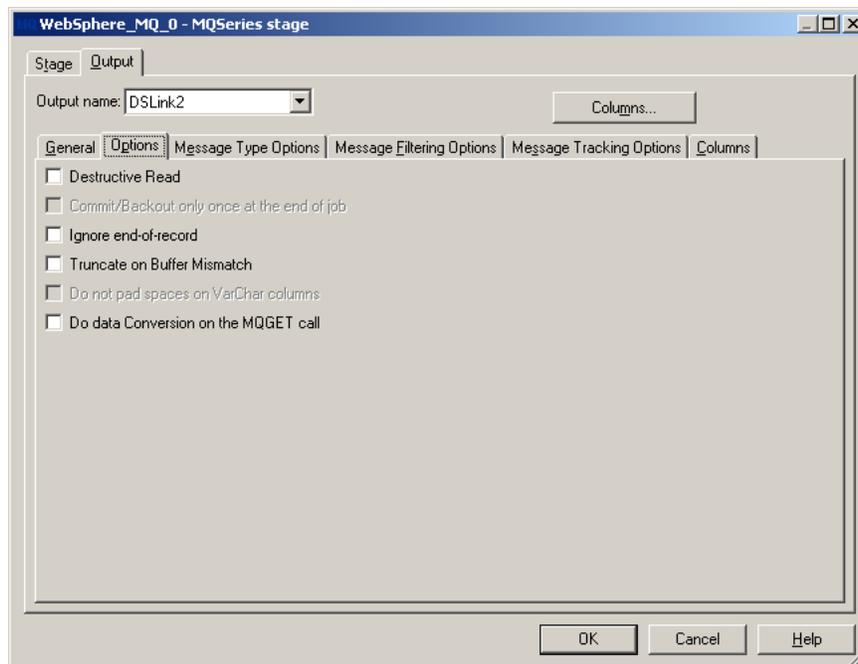
- **End of Data Message.** An integer, which is stored in the message descriptor, that represents the user-defined message type. A default value of 0 indicates that no special messages should be expected to signal the termination of message reading. Typically, the stage processes datagram or request messages. If the stage receives a message whose type matches this integer, it ends the processing.
- **Description.** Optional. A description of the output link.

Note A job can run indefinitely if it does not reach **Message Limit** or if it does not receive **End of Data Message**. To avoid this possibility, specify a suitable value for **Wait Time** on the **General** tab of the **Output** page.

For more troubleshooting information, see "[Troubleshooting](#)" on page 10.

Options Tab

This tab provides options determining how messages are read.



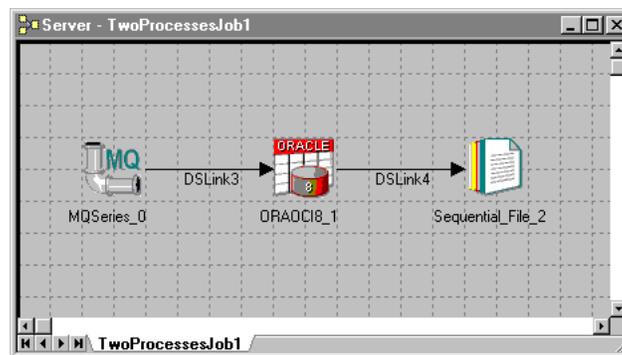
The **Options** tab contains the following check boxes:

- **Destructive Read.** The type of read to be executed. If not selected (the default), the message is left on the queue after it is read. If selected, each message is deleted from the queue immediately after it is read.

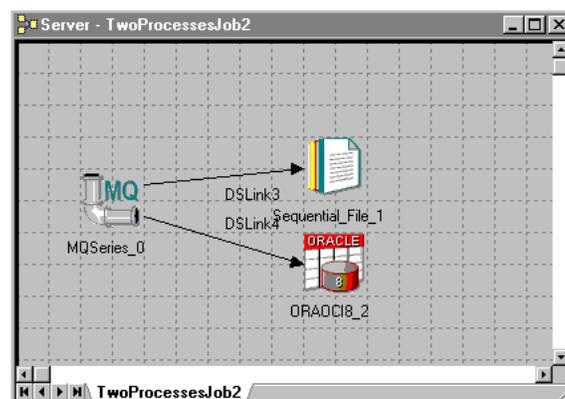
Note Do not use **Destructive Read** in a job with local unit of work (see "[Processing Units of Work](#)" on page 7).

- Commit/Backout only once at end of job.** A further refinement of the type of read to be executed. This check box is active only if **Destructive Read** is selected. If **Commit/ Backout only once at end of job** is selected, messages are deleted only when the job finishes successfully. If the job does not finish successfully, messages remain on the queue. If not selected (the default), each message is deleted from the queue immediately after it is read.

Do not use **Commit/Backout only once at end of job** with any job consisting of multiple processes. In the following example, the job writes to a table or a file and reads from that table or file, Ascential DataStage introduces multiple processes in this configuration.



In the next example, there are two target files directly linked to the WebSphere MQ stage, and again Ascential DataStage introduces multiple processes.

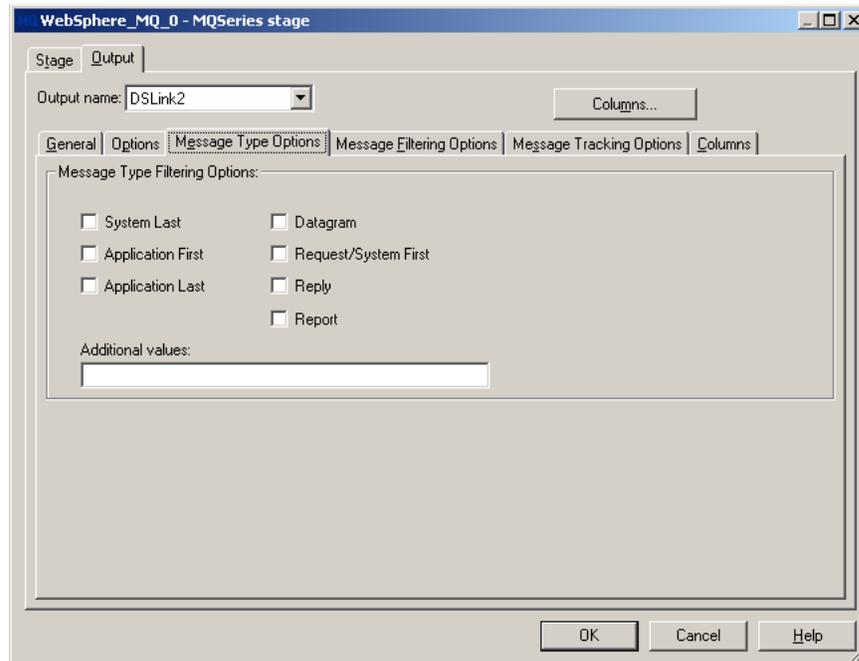


Note If an After SQL statement fails and the job is aborted, messages are removed from the queue even if **Commit/Backout only once at end of job** is selected, because the message has already been successfully moved to the target.

- **Ignore end-of-record.** A check box specifying that the stage does not treat carriage returns or new lines in the message stream as the end of a logical DataStage row. If cleared, a carriage return or new line indicates the end of a logical DataStage row. Therefore, a single WebSphere MQ message can result in many DataStage rows.
- **Truncate on Buffer Mismatch.** A check box specifying whether to truncate a message. If selected and the buffer size is smaller than an incoming message, the stage truncates the message. If selected and the buffer size is greater than an incoming message, the stage pads the message with blanks or zeroes. If not selected (the default) and the buffer size is smaller than an incoming message, the stage splits the message into several rows.
- **Do not pad spaces on VarChar columns.** The treatment of a message when its length is less than the maximum length of the VarChar column. If selected, the column is not padded with spaces at the end of the message. **Do not pad spaces on VarChar columns** is not active unless **Truncate on Buffer Mismatch** is selected.
- **Do data Conversion on the MQGET call.** A check box specifying automatic conversion is to take place. If the check box is selected, the stage sets the MQGMO_CONVERT option in the Message Get Options structure, and automatic conversion occurs. If the check box is not selected (the default), the stage does not set the MQGMO_CONVERT option, and conversion does not occur.

Message Type Options Tab

This tab provides options to limit which messages are read.



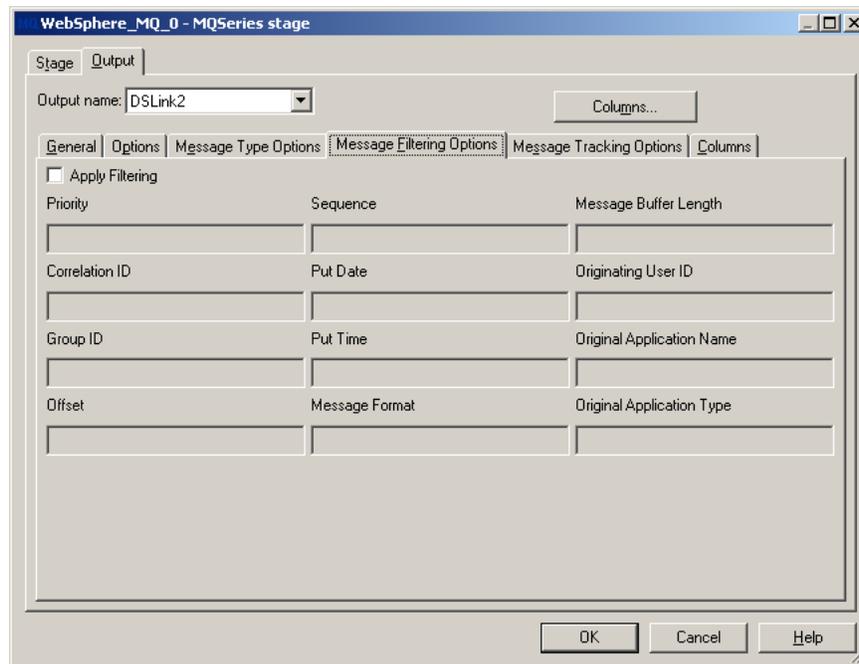
The **Message Type Options** tab contains a set of check boxes and a place to specify additional values. Only messages with a message type matching the selected message filter options are extracted from the queue and passed as columns on the output link. **System First** and **System Last** together make up a range for system-specific values. **Application First** and **Application Last** together make up a range for application-specific values.

Use **Additional values** to provide specific values. You can provide numbers and/or ranges of numbers separated by commas.

The values indicated by the check boxes and the values provided in **Additional values** are concatenated into one comma-delimited string value for processing.

Message Filtering Options Tab

This tab provides options to filter or validate messages. Every input value must meet the specified condition.



The **Message Filtering Options** tab contains the following fields:

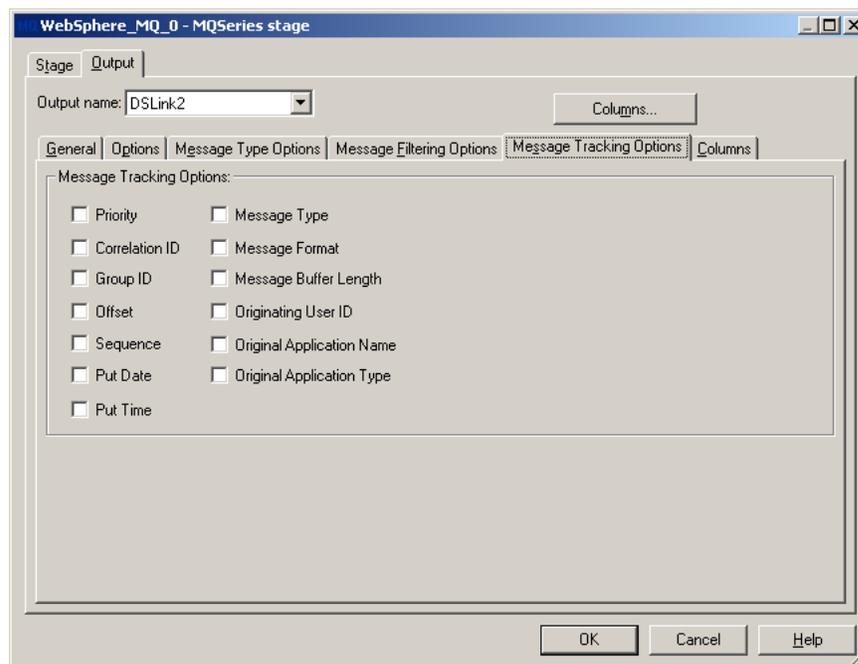
- **Apply Filtering.** This check box specifies whether filtering is active. If not selected (the default), filtering is not active.
- The following table contains a list of properties that can be filtered and the condition against which each is validated:

Property	Condition against which Validated
Priority	Value from 0 to 999,999,999
Correlation ID	Character string limited to 24 characters for each element that is not a job parameter
Group ID	Character string limited to 24 characters for each element that is not a job parameter
Offset	Value from 0 to 999,999,999
Sequence	Value from -1 to 999,999,999
Put Date	YYMMDD (limited to 8 characters for each element if not a job parameter)
Put Time	HHMMSS (limited to 8 characters for each element if not a job parameter)

Property	Condition against which Validated
Message Format	Character string limited to 8 characters for each element that is not a job parameter
Message Buffer Length	Value from 0 to 4194304
Originating User ID	Character string limited to 12 characters for each element that is not a job parameter
Original Application Name	Character string limited to 28 characters for each element that is not a job parameter
Original Application Type	Value from 1 to 999999999

Message Tracking Options Tab

This tab provides options to track messages.



The **Message Tracking Options** tab contains a subset of the message tracking data that can be extracted from the message descriptor and passed as columns on the output link. The message tracking data include:

- **Priority.** The message priority
- **Correlation ID.** The message correlation identifier
- **Group ID.** The message group identifier
- **Offset.** The message offset for segmented messages

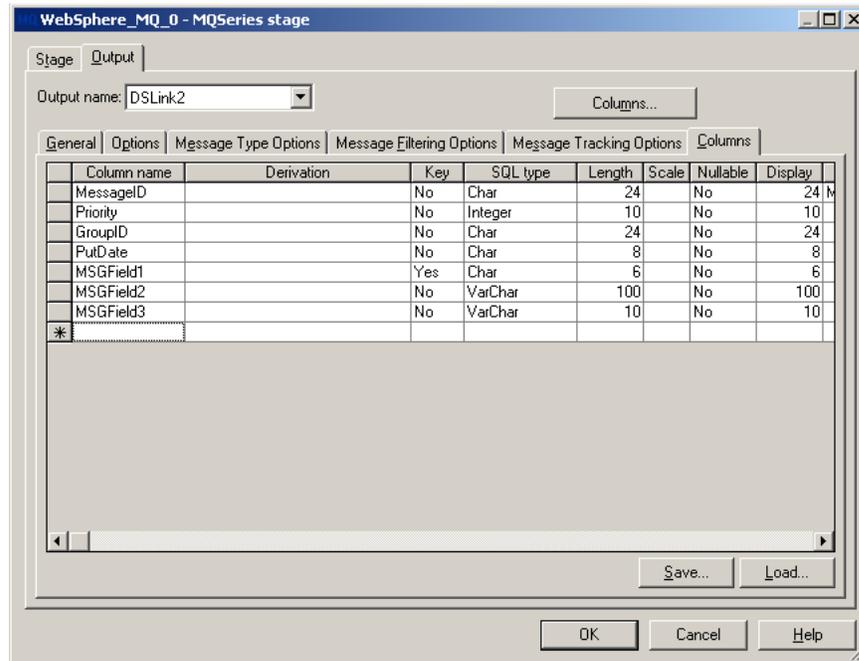
- **Sequence.** The message sequence number
- **Put Date.** The unchanged message put date
- **Put Time.** The unchanged put time
- **Message Type.** The message type
- **Message Format.** The format name of the message data
- **Message Buffer Length.** The length of the message returned from the read
- **Originating User ID.** The user identifier
- **Original Application Name.** The name of the application that put the message
- **Original Application Type.** The type of application that put the message

The column meta data for these message tracking items is automatically maintained by the user interface, as described in the section "[Columns Tab](#)"

Columns Tab

This tab contains the column definitions for the data being output on the chosen link. For a general description of how to enter and edit column definitions, see *Ascential DataStage Designer Guide*. For information about the data elements in the WebSphere MQ stage, see "[Using Column Data Elements](#)" on page 33.

The following example demonstrates how the selection of message tracking options affects the column definitions on the output link:



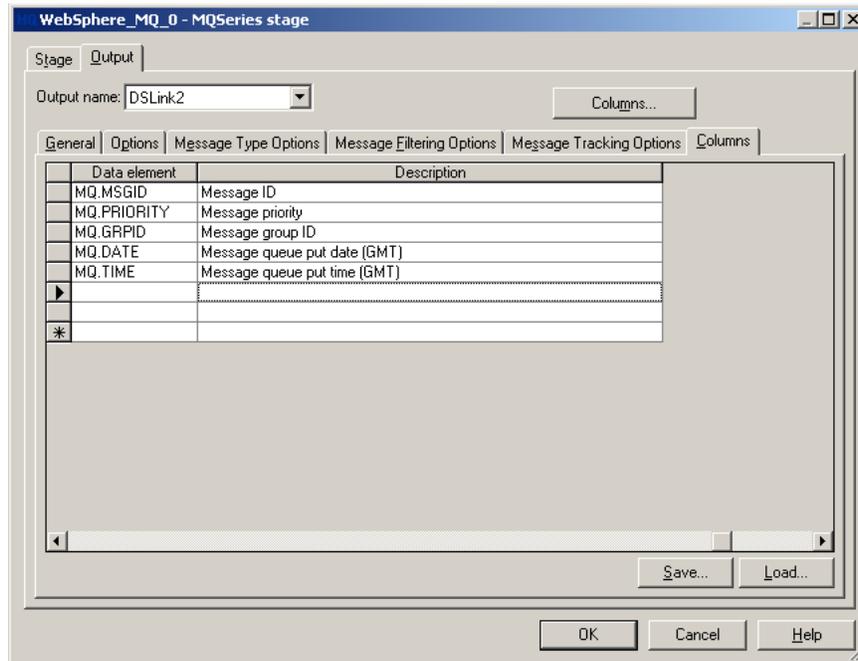
The first column name of an output link is always the **Message ID**. This column is used in transactional situations to do syncpoint reads from the source queue. For information, see ["Processing Units of Work"](#) on page 7.

In this same example, the four column names following the **Message ID** represent the message tracking options selected on the **Message Tracking Options** tab for the **Output** page. You can modify these column names to avoid column-naming conflicts.

We recommend that you do not modify:

- The data element type that the WebSphere MQ stage uses to identify these special columns. However, you can select alternative data element types for the **PutDate** and **PutTime** tracking options. For a description of the data elements that represent the various message tracking options, see ["Using Column Data Elements"](#) on page 33.
- SQL type. If the stage detects at run-time that the column definitions do not correspond to the selected message tracking options, the job aborts.

The following screen shows the **Data element** field for the columns representing the **Message Tracking Options**:



The remaining columns (MSGField1 and so forth) represent the logical column names of the fields contained in the WebSphere MQ message. A WebSphere MQ message is a string-formatted message of fixed-length records, except the last column, which can be shorter. Therefore, you should specify character data types and lengths for these columns.

Note the following rules about the length of the data for columns on the **Input** and **Output** pages:

- If data for a column is shorter than the column width as specified by the **Display** value, the data is padded with trailing spaces if defined as Char. For VarChar data, **Do not pad spaces on VarChar** takes precedence.
- If data is longer than the column width as specified by the **Display** value, it is truncated to the specified column width (**Input** only).
- If the column width as specified by the **Display** value is empty, the width is determined based on the **SQL type** and the data precision (specified by the **Length** value) as well as on whether **Buffer Mismatch** (on the **Output** page) or **Do not pad spaces on VarChar** (on the **Input** page) is selected.

The following list documents the SQL data types and the specifications for their column widths:

- BigInt, Integer, SmallInt, TinyInt. The column width is the Length plus 1 for the optional sign.

- **Numeric, Decimal.** The column width is the Length plus 2 for the optional sign and decimal point
- **Float, Real, Double.** The column width is the Length plus 7 for the optional sign, decimal, and exponential expressions, for example, -1.2000e-009.
- **Date.** The column width is 10, using the *YYYY-MM-DD* format.
- **Time.** The column width is 8, using the 24-hour *HH:MM:SS* format.
- **Timestamp.** The column width is 19, using the *YYYY-MM-DD HH:MM:SS* 24-hour format.
- **Others.** (such as Unknown, Char, VarChar, LongVarChar, NChar, NVarChar, LongNVarChar, Binary, VarBinary, and LongVarBinary, Bit) These equal the Length.

Since dates and times are character strings, when messages containing dates or times are read from WebSphere MQ into Ascential DataStage or vice versa, the DataStage dates and times are in internal format.

Using Column Data Elements

In addition to the message ID, you can choose other message descriptor fields for delivery on the links. The column definitions representing these message properties appear before any columns representing actual message data.

The queue manager generates these message descriptor properties, except **Priority**, in a format that may be meaningless to other downstream stages.

For example, the message **PutTime** is expressed as *HHMMSSSTH* where *T* represents tenths of a second and *H* represents hundredths of a second. You may want to express this value as *HH:MM:SS.TH* or in internal DataStage time format. **PutDate** is expressed as *YYYYMMDD*.

The following data elements specify transformations on message put dates and times. The WebSphere MQ stage performs these transformations, so a separate Transformer stage is unnecessary.

- **MQ.DATE.TO.TAG.** Converts *YYYYMMDD* to *YYYY-MM-DD*.
- **MQ.DATE.TO.DSDATE.** Converts *YYYYMMDD* to internal DataStage date (days since 12/31/67).
- **MQ.TIME.TO.TAG.** Converts *HHMMSSSTH* to *HH:MM:SS.TH*.

- **MQ.TIME.TO.DSTIME.** Converts *HHMMSSSTH* to internal DataStage time.

The following data elements associate the message descriptor properties with their corresponding output columns. They are used only for identification purposes, not to imply any transformations on the message data. The stage uses these elements to identify output columns associated with message descriptors. Therefore, you can modify the column names in your job design, if necessary.

- **MQ.MSGID.** The message ID, which is a mandatory output column.
- **MQ.PRIORITY.** The message priority.
- **MQ.CORRID.** The message correlation ID.
- **MQ.GRPID.** The message group ID.
- **MQ.OFFSET.** The message offset for segmented messages.
- **MQ.SEQUENCE.** The message sequence number.
- **MQ.DATE.** The unchanged message put date.
- **MQ.TIME.** The unchanged put time.
- **MQ.MSGTYPE.** The message type
- **MQ.MSGTYPE.TO.STR.** The message type converted to a string value
- **MQ.MSGFORMAT.** The format name of the message data
- **MQ.MSGBUFLEN.** The length of the message returned from the read
- **MQ.PUTUSERID.** The user identifier
- **MQ.PUTAPPLNAME.** The name of the application that put the message
- **MQ.PUTAPPLTYPE.** The type of application that put the message
- **MQ.PUTAPPLTYPE.TO.STR.** The type of application that put the message converted to a string value

Note Columns for the Message ID, Correlation ID and Group ID data elements contain binary data that is not NLS-mappable. If this data is written to a non-MQ Series stage, we recommend turning NLS *off* on a per-column basis for these columns in the non-MQ Series stages (set **NLS Map** to **NONE** on the **Columns** tab of the **Input** page) after selecting the **Allow per-column mapping** box on the **NLS** tab.

Index

B

browse mode 9
browsing the queue 4, 7

C

cluster 10
column definitions 6
Columns tab
 Input page 19
 Output page 30
complex message formats 5
configuration requirements 4
configuration requirements, Linux 4
connection parameters 14

D

destructive read 24

E

examples 8
executing a queue commit 9

F

filtering 28
functionality 2

G

General tab
 Input page 16
 Output page 22
 Stage page 14

I

infinite loops 10
Input page 13, 15
 Columns tab 19
 General tab 16
 Options tab 18
 Transaction Handling tab 19
installation 4

L

length of data, rules 32
link execution, order of 15
Linux, configuration requirements 4

M

message descriptor fields 33
message details 5
message expiry 16
Message Filtering Options tab 28
message limit 10, 23
message persistence 17
message priority 16
message queues 5
Message Tracking Options tab 29
message type 16
Message Type Options tab 27
messages 6

N

namelists 5
NLS tab 13, 14

O

Options tab
 Input page 18
 Output page 24
order of link execution 15

Output page 13, 22
 Columns tab 30
 General tab 22
 Message Filtering Options tab 28
 Message Tracking Options tab 29
 Message Type Options tab 27
 Options tab 24

P

permissions 12
persistnece of message 17
priority of message 16

Q

queue commit 9
queue details 5
queue manager 5, 14
queue manager configuration 10
queue name 5

R

remote queue definition 10
reply and report messages 11
row length, defining 6
rows in a message 6
rules, length of data 32

S

security 11
setmqaut utility 12
Stage page 13
 General tab 14
 NLS tab 13, 14
staging area 10, 11

T

terminology 3
transaction control 4, 20
transaction groups 20
Transaction Handling tab
 Input page 19
troubleshooting 10
type of message 16

U

unit of work 4, 7

W

wait time 23
writing messages 9

X

XML format 6