

# Ascential **DataStage**

---

## Administrator Guide

Version 7.5.1



his 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.

This product or the use thereof may be covered by or is licensed under one or more of the following issued patents: US6604110, US5727158, US5909681, US5995980, US6272449, US6289474, US6311265, US6330008, US6347310, US6415286; Australian Patent No. 704678; Canadian Patent No. 2205660; European Patent No. 799450; Japanese Patent No. 11500247.

© 2005 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.

The software delivered to Licensee may contain third-party software code. See [Legal Notices](#) (LegalNotices.pdf) for more information.

# How to Use this Guide

Ascential DataStage™ is a tool set for designing, developing, and running jobs that populate tables in a data warehouse or data mart.

This manual describes DataStage administration, including:

- How to use the DataStage Administrator
- Essential system administration tasks that must be performed through the operating system
- Other housekeeping tasks, such as customizing menus and setting defaults

To find particular topics in the guide, you can:

- Use the Guide's contents list (at the beginning of the Guide).
- Use the Guide's index (at the end of the Guide).
- Use the Adobe Acrobat Reader bookmarks.
- Use the Adobe Acrobat Reader search facility (select **Edit** ► **Search**).

The guide contains links both to other topics within the guide, and to other guides in the DataStage manual set. The links are shown in [blue](#). Note that, if you follow a link to another manual, you will jump to that manual and lose your place in this manual. Such links are shown in *italics*.

## Organization of This Manual

This manual contains the following:

- [Chapter 1](#) describes how to configure DataStage® to suit the needs of your system.
- [Chapter 2](#) describes DataStage administration on a Windows server.
- [Chapter 3](#) describes DataStage administration on a UNIX server.
- [Chapter 4](#) describes how to use National Language Support (NLS) with DataStage.

- **Chapter 5** outlines troubleshooting procedures to solve possible problems.

## Documentation Conventions

This manual uses the following conventions:

Convention	Usage
<b>Bold</b>	In syntax, bold indicates commands, function names, keywords, and options that must be input exactly as shown. In text, bold indicates keys to press, function names, and menu selections.
UPPERCASE	In syntax, uppercase indicates BASIC statements and functions and SQL statements and keywords.
<i>Italic</i>	In syntax, italic indicates information that you supply. In text, italic also indicates UNIX commands and options, file names, and pathnames.
Plain	In text, plain indicates Windows commands and options, file names, and path names.
Lucida Typewriter	The Lucida Typewriter font indicates examples of source code and system output.
<b>Lucida Typewriter</b>	In examples, Lucida Typewriter bold indicates characters that the user types or keys the user presses (for example, <b>&lt;Return&gt;</b> ).
[ ]	Brackets enclose optional items. Do not type the brackets unless indicated.
{ }	Braces enclose nonoptional items from which you must select at least one. Do not type the braces.
itemA   itemB	A vertical bar separating items indicates that you can choose only one item. Do not type the vertical bar.
...	Three periods indicate that more of the same type of item can optionally follow.
➤	A right arrow between menu commands indicates you should choose each command in sequence. For example, "Choose <b>File</b> ➤ <b>Exit</b> " means you should choose <b>File</b> from the menu bar, then choose <b>Exit</b> from the File pull-down menu.
This line ➤ continues	The continuation character is used in source code examples to indicate a line that is too long to fit on the page, but must be entered as a single line on screen.

The following are also used:

- Syntax definitions and examples are indented for ease in reading.

- All punctuation marks included in the syntax—for example, commas, parentheses, or quotation marks—are required unless otherwise indicated.
- Syntax lines that do not fit on one line in this manual are continued on subsequent lines. The continuation lines are indented. When entering syntax, type the entire syntax entry, including the continuation lines, on the same input line.

## DataStage Documentation

DataStage documentation includes the following:

- **DataStage Administrator Guide:** This guide describes DataStage setup, routine housekeeping, and administration.
- **DataStage Designer Guide:** This guide describes the DataStage Designer, and gives a general description of how to create, design, and develop a DataStage application.
- **DataStage Manager Guide:** This guide describes the DataStage Manager and describes how to use and maintain the DataStage Repository.
- **DataStage Server: Server Job Developer's Guide:** This guide describes the tools that are used in building a server job, and it supplies programmer's reference information.
- **DataStage Enterprise Edition: Parallel Job Developer's Guide:** This guide describes the tools that are used in building a parallel job, and it supplies programmer's reference information.
- **DataStage Enterprise Edition: Parallel Job Advanced Developer's Guide:** This guide gives more specialized information about parallel job design.
- **DataStage Enterprise MVS Edition: Mainframe Job Developer's Guide:** This guide describes the tools that are used in building a mainframe job, and it supplies programmer's reference information.
- **DataStage Director Guide:** This guide describes the DataStage Director and how to validate, schedule, run, and monitor DataStage server jobs.
- **DataStage Install and Upgrade Guide.** This guide contains instructions for installing DataStage on Windows and UNIX platforms, and for upgrading existing installations of DataStage.
- **DataStage NLS Guide.** This Guide contains information about using the NLS features that are available in DataStage when NLS is installed.

These guides are also available online in PDF format. You can read them using the Adobe Acrobat Reader supplied with DataStage. See *Install and Upgrade Guide* for details on installing the manuals and the Adobe Acrobat Reader.

You can use the Acrobat search facilities to search the whole DataStage document set. To use this feature, select **Edit ► Search** then choose the **All PDF documents in** option and specify the DataStage docs directory (by default this is C:\Program Files\Ascential\DataStage\Docs).

Extensive online help is also supplied. This is particularly useful when you have become familiar with DataStage, and need to look up specific information.

# Contents

## How to Use this Guide

Organization of This Manual .....	iii
Documentation Conventions .....	iv
DataStage Documentation .....	v

## Chapter 1 Configuring DataStage

Who Can Administer DataStage? .....	1-1
The DataStage Administrator .....	1-2
Server Properties .....	1-4
Setting the Server Timeout .....	1-4
DataStage Project Administration .....	1-5
Adding Projects .....	1-5
Deleting Projects .....	1-7
Moving Projects .....	1-7
Project Properties .....	1-9
General Page .....	1-10
Permissions Page .....	1-15
Enabling Tracing on the Server .....	1-17
Schedule Page .....	1-19
Mainframe Page .....	1-20
Tunables Page .....	1-23
Parallel Page .....	1-25
Sequence Page .....	1-26
Remote Page .....	1-26

DataStage License Administration . . . . .	1-31
Changing Server License Details . . . . .	1-31
Changing Client License Details . . . . .	1-32
Issuing DataStage Engine Commands. . . . .	1-35
Customizing the Tools Menu . . . . .	1-37
Adding a Tool . . . . .	1-37
Changing the Tools Menu Order . . . . .	1-38
Adding a Separator . . . . .	1-38
DataStage NLS. . . . .	1-39

## **Chapter 2**

### **Windows Server Administration**

User Roles on Windows Systems . . . . .	2-1
Starting the Schedule Service . . . . .	2-2
Stopping and Restarting the Server Engine . . . . .	2-2
Accessing ODBC Databases . . . . .	2-3
Accessing UniVerse Systems . . . . .	2-4
Installing DataStage Packages . . . . .	2-5
Package Installer Command Line Options . . . . .	2-6
Installing MetaBrokers. . . . .	2-7

## **Chapter 3**

### **UNIX Server Administration**

Installation Requirements . . . . .	3-2
User Roles on UNIX Systems . . . . .	3-2
Limiting Access to Projects . . . . .	3-3
Tuning UNIX for DataStage . . . . .	3-4
Running Out of File Units . . . . .	3-4
The DataStage Server Daemon (dsrpcd) . . . . .	3-5
Environment Requirements for dsrpcd . . . . .	3-5
Stopping and Restarting the Server Engine . . . . .	3-6
The DataStage Environment on UNIX. . . . .	3-6
Accessing UniVerse Systems . . . . .	3-7
Accessing ODBC Databases . . . . .	3-9
Adding Data Sources to uvodbc.config . . . . .	3-9
Installing DataStage Packages . . . . .	3-10

Server Restrictions . . . . .	3-10
Connecting to the Server . . . . .	3-10
Scheduling Jobs . . . . .	3-11
Accessing Remote Files via NFS. . . . .	3-11

**Chapter 4**  
**NLS Configuration**

Changing Project Maps . . . . .	4-2
Server Job Project Maps . . . . .	4-2
Parallel Job Project maps . . . . .	4-3
Changing Project Locales . . . . .	4-4
Server Job Locales . . . . .	4-4
Parallel Job Locales . . . . .	4-6
Client/Server Maps. . . . .	4-6

**Chapter 5**  
**Troubleshooting**

Error Messages . . . . .	5-1
Scheduled Jobs . . . . .	5-2
Windows Scheduling Problems . . . . .	5-2
UNIX Scheduling Problems. . . . .	5-4
Job Termination Problems . . . . .	5-5
ODBC Connection Problems on UNIX . . . . .	5-5
Check the Shared Library Environment . . . . .	5-6
Check Symbolic Links . . . . .	5-6
DataStage Client to UNIX Server Connections . . . . .	5-6
Connecting to UniData Data Sources. . . . .	5-7



# 1

## Configuring DataStage

This chapter describes how to set up DataStage when it is first installed, and some routine administration tasks, including:

- Changing license details
- Setting up DataStage users
- Adding, deleting, and moving DataStage projects
- Purging job log files
- Setting the timeout interval on the server computer
- Tracing server activity
- Adding entries to the **Tools** menu
- Setting job parameter defaults
- Issuing DataStage Engine commands from the Administration client

### Who Can Administer DataStage?

You can do many of the administration tasks described in this chapter if you have been defined as a DataStage Developer or a DataStage Production Manager. You do not need to have specific administration rights. However, to do some tasks you must be logged on to DataStage using a user name that gives you administrator status:

- *For Windows servers:* You must be logged on as a member of the Windows Administrators group.
- *For UNIX servers:* You must be logged in as *root* or the DataStage administrative user (*dsadm* by default).

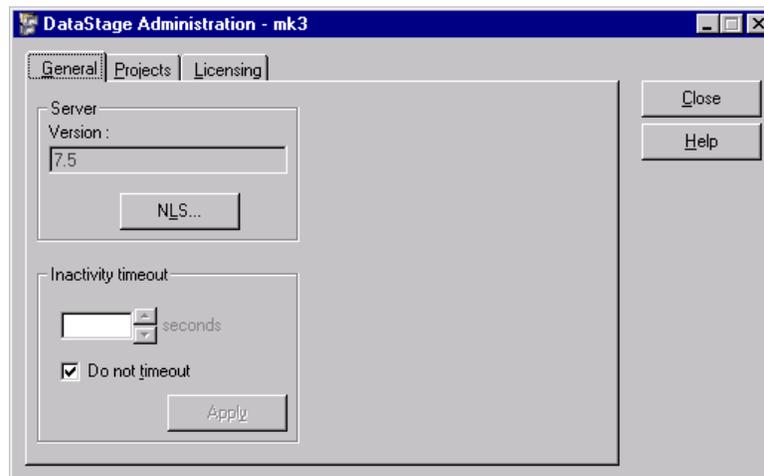
You require administrator status to, for example, change license details, add and delete projects, or to set user group assignments. The descriptions of the administration procedures in this manual state when administrator status is required.

## The DataStage Administrator

Most DataStage configuration tasks are carried out using the DataStage Administrator, a client program provided with DataStage. To access the DataStage Administrator:

- 1 From the Ascential DataStage program folder, choose **DataStage Administrator**.
- 2 Log on to the server. If you do so as an Administrator (for Windows servers), or as *root* or the specified administrative user (for UNIX servers), you have unlimited administrative rights; otherwise your rights are restricted as described in the previous section.

The DataStage Administration window appears:

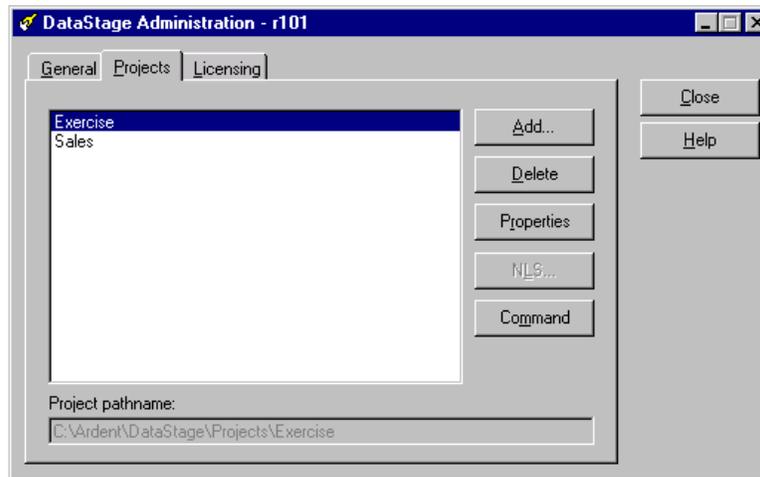


The **General** page lets you set server-wide properties. It is enabled only when at least one project exists.

The controls and buttons on this page are enabled only if you logged on as an administrator (see "[Who Can Administer DataStage?](#)" on page 1-1). In addition, you can use the **NLS...** button only when DataStage NLS (National Language Support) is enabled.

For further information about setting server-wide properties, refer to [page 1-4](#).

To bring the **Projects** page to the front, click the **Projects** tab. This page lists the DataStage projects, and shows the pathname of the selected project in the **Project pathname** field.



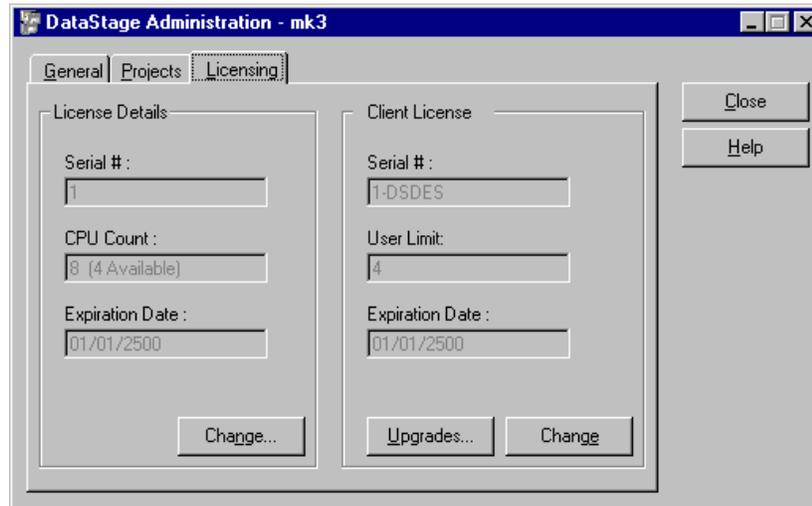
The **Projects** page has the following buttons:

- **Add...** adds new DataStage projects. This button is enabled only if you have administrator status (see "[Who Can Administer DataStage?](#)" on page 1-1).
- **Delete** deletes projects. This button is enabled only if you have administrator status.
- **Properties** sets the properties of the selected project.
- **NLS...** lets you change project maps and locales.
- **Command** issues DataStage Engine commands directly from the selected project.

If no projects exist, only the **Add...** and **Command** buttons are enabled, and **Command** issues DataStage Engine commands on the DataStage Engine account.

The **Licensing** page allows you to view the current details for your server and client licenses, and upgrade these without the need to

reinstall. You can upgrade a license to increase user limit and expiration date.



## Server Properties

The **General** page on the DataStage Administration window (see [page 1-2](#)) lets a user with administrator status set the following server-wide properties:

- The server timeout
- The NLS client/server map

**Note** The controls and buttons on this page are enabled only if you have administrator status (see "[Who Can Administer DataStage?](#)" on page 1-1).

For information about NLS client/server maps, and how to change them, refer to "[Client/Server Maps](#)" on [page 4-6](#).

## Setting the Server Timeout

By default, the connection between the DataStage client and server times out after 86400 seconds (24 hours) inactivity. To change the default:

- 1 In the DataStage Administration window, click the **General** tab to move the **General** page to the front.
- 2 In the **Inactivity Timeout** area, use the up and down buttons to change the timeout period, or enter the new timeout period in the **seconds** field.

- 3 To disable inactivity timeout, select the **Do not timeout** check box.
- 4 Click **Apply** to apply the new settings. The changes take effect when you restart the server engine.

If inactivity timeout is already disabled when the DataStage Administrator starts, the timeout reverts to the 86400-second default when you reenables it.

The timeout also affects the `dsjob -wait` command issued from the command line and the job control api function `DSWaitForJob`. If you are using these features you are advised to set a long timeout, or disable the timeout altogether. It also affects Shared Containers on the Parallel canvas (which use `DSWaitForJob`).

## DataStage Project Administration

This section describes how to do the following:

- Add new DataStage projects
- Delete projects
- Move projects

When you delete DataStage projects, always do so through the DataStage Administrator to ensure that licensing information is updated correctly.

You must have Administrator status in order to add or delete projects (see "[Who Can Administer DataStage?](#)" on page 1-1).

### Adding Projects

You can add further projects from the DataStage Administrator as required.

You can also specify that a new project is protected. This is a special category of project and, normally, nothing can be added, deleted or changed in the project. Users can view objects in the project, and perform tasks that affect the way a job runs rather than the jobs design; specifically they can:

- Run jobs
- Set job properties
- Set job parameter default values

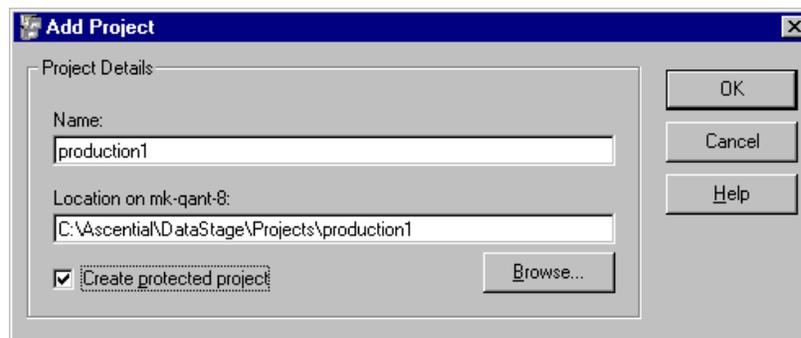
A newly created protected project is populated by importing developed jobs and components; only a Production Manager user can

perform the import, no other types of user can import into a special project. (To mark an existing project as protected, use the **Project Properties** dialog box – see [page 1-9](#)).

Protected projects provide a way of maintaining the integrity of jobs intended to be run in a 'production' environment, i.e., the finished jobs. Giving widespread access to the Production Manager role would obviate the benefits of using protected projects. See "[User Roles on Windows Systems](#)" on [page 2-1](#) and "[User Roles on UNIX Systems](#)" on [page 3-2](#) for advice on setting up user roles.

To add a new project:

- 1 Click the **Projects** tab on the DataStage Administration window (see [page 1-2](#)) to move this page to the front.
- 2 Click the **Add...** button. The **Add Project** dialog box appears:



**Note** The Add... button is enabled only if you have administrator status (see "[Who Can Administer DataStage?](#)" on [page 1-1](#)).

- 3 Enter the project name in the **Name** field. It is automatically appended to the default location for the new project.  
**Note** You cannot add a project to a location that already exists.
- 4 To change the default location, enter a new location in the **Location on server** field, or use the **Browse...** button to select a new location.
- 5 If you want the project to be a protected one, select the **Create protected project** check box.
- 6 Click **OK**.

You can also add projects by repeating the server installation from the DataStage Master Setup screen. If the installed version of DataStage matches the release on the distribution CD-ROM, the installation program automatically enters maintenance mode, giving you the option to add projects to the current installation.

## Deleting Projects

You must have administrator status in order to delete a project (see ["Who Can Administer DataStage?"](#) on [page 1-1](#)).

To delete a project from the server:

- 1** Ensure that no users are connected to the project you want to delete. DataStage returns an error message if you try to delete a project while another user is connected to it.
- 2** Click the **Projects** tab on the [DataStage Administration window](#) (see [page 1-2](#)) to move this page to the front.
- 3** Select the project you want to delete, then click **Delete**.
- 4** You are prompted to confirm the deletion. If you confirm, the project is deleted immediately.

## Moving Projects

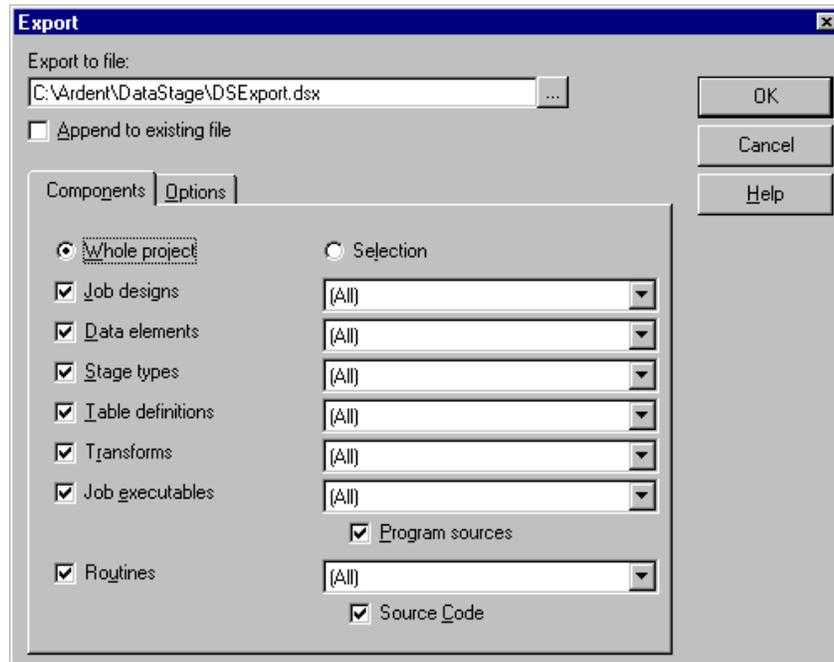
To move a DataStage project, you must export it to a file and then import it to the new location. You then need to delete the original project using the Administrator client.

**Note** This procedure can be used only on systems that have the DataStage Developer's Edition installed.

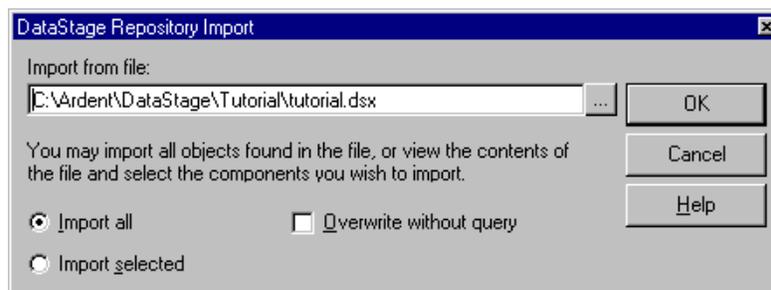
To move a project:

- 1** Start the DataStage Director in the project you want to move, and check that no jobs are running.
- 2** Choose **Tools ► Run Manager...** . The DataStage Manager window appears.
- 3** Close the DataStage Director to disconnect from the project.

- 4 Choose **Export ► DataStage Components...** . The **Export** dialog box appears:



- 5 Enter a file name to hold the exported project.
- 6 Click the **Whole project** option button.
- 7 Click **Export**. The project is exported to the file.
- 8 Click **Close** to close the **Export** dialog box.
- 9 Add a new project in the required location (see ["Adding Projects"](#) on [page 1-5](#)).
- 10 Open the DataStage Manager in the new project (note that, if the new project is protected, then you need to be a Production Manager user in order to perform the import).
- 11 Choose **Import ► DataStage Components...** . The **DataStage Repository Import** dialog box appears:



- 12 Enter the file name you used for the exported project in step 5.

- 13** Click the **Import all** option, then click **OK**. The project is imported to the new location.

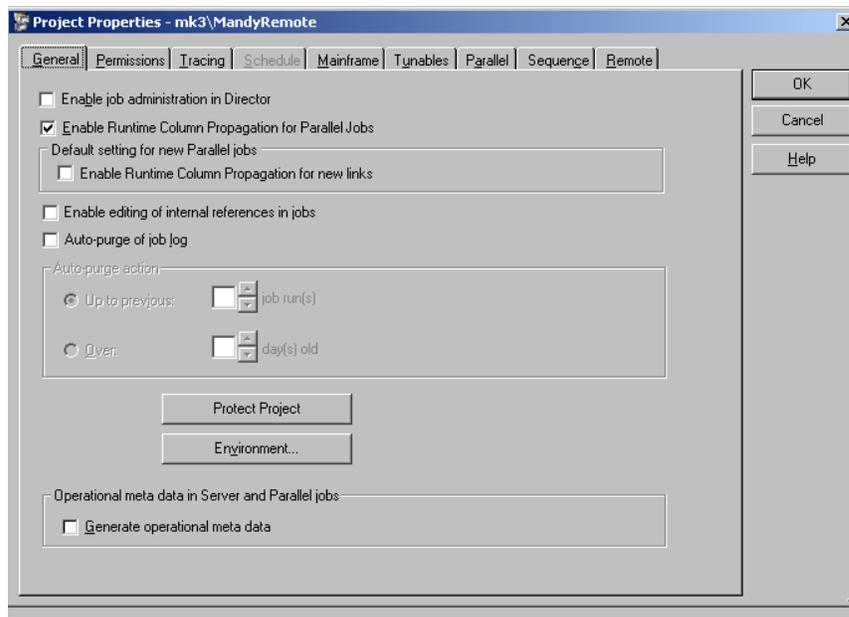
**Note** Mainframe-specific objects in the import file are imported only if the destination project supports mainframe jobs.

When you move projects, project level settings such as environment variable settings and project options are not included.

## Project Properties

To view and edit the properties of a project:

- 1** Click the **Projects** tab on the [DataStage Administration window](#) (see [page 1-2](#)) to move the **Projects** page to the front.
- 2** Select the project.
- 3** Click the **Properties** button. The Project Properties window appears:



You use the pages in the Project Properties window to do the following:

- **General.** Enable job administration from the DataStage Director, enable runtime column propagation for parallel jobs, define a project-wide setting for auto-purge of the job log, and set up environment variables. If MetaStage is installed, you can also configure DataStage to send it meta data generated by project jobs.

- **Permissions.** Assign user categories to operating system user groups, or enable operators to view all the details of an event in a job log file. The **Permissions** tab is enabled only if you have logged on to DataStage using a name that gives you administrator status.
- **Tracing.** Enable or disable tracing on the server.
- **Schedule.** Set up a user name and password to use for running scheduled DataStage jobs. The **Schedule** tab is enabled only if you have logged on to a Windows server.
- **Mainframe.** Set mainframe job properties and the default platform type. This page is enabled only if XE/390 is installed.
- **Tunables.** Configure cache settings for Hashed File stages.
- **Parallel.** Set parallel job properties and defaults for date/time and number formats.
- **Sequence.** Set compilation defaults for job sequences.
- **Remote.** If you have specified that parallel jobs in the project are to be deployed on a USS system, this page allows you to specify deployment mode and USS machine details.

## General Page

### Enabling Job Administration in the DataStage Director

From the Administration client, the administrator can enable or disable job administration features in the DataStage Director. They let DataStage operators release the resources of a job that has aborted or hung, and so return the job to a state in which it can be rerun when the cause of the problem has been fixed. The default is for these features to be disabled.

To enable job administration in the DataStage Director:

- 1 Click the **Projects** tab in the [DataStage Administration window](#) (see [page 1-2](#)) to move this page to the front.
- 2 Select the project.
- 3 Click **Properties**. The Project Properties window appears, with the **General** page displayed as shown on [page 1-9](#).
- 4 Select the **Enable job administration in Director** check box.
- 5 Click **OK**.

This procedure enables two commands in the Director **Job** menu:

- **Cleanup Resources**

### ■ **Clear Status File**

**Cleanup Resources** is also enabled in the Monitor window shortcut menu in the Director. **Cleanup Resources** lets the user:

- View and end job processes
- View and release the associated locks

**Note** **Cleanup Resources** is not compatible with server tracing (see ["Enabling Tracing on the Server"](#) on [page 1-17](#)). If you enable server tracing and job administration, the DataStage Director displays an error message when **Cleanup Resources** is chosen.

**Clear Status File** removes the status records associated with all stages of the selected job. It should be used only if the user believes all the job processes have gone away and the job cannot be reset correctly.

**Warning** These two commands give the user considerable control over job resources, and should be used with care.

Refer to ["Cleaning Up Job Resources"](#) in *DataStage Director Guide* for full information about how to use these commands.

## **Enable Runtime Column Propagation for Parallel Jobs**

If you enable this feature, stages in parallel jobs can handle undefined columns that they encounter when the job is run, and propagate these columns through to the rest of the job. This check box enables the feature, to actually use it you need to explicitly select the option on each stage (see [Parallel Job Developer's Guide](#)). Enabling this feature makes the following sub-property available:

- **Enable Runtime Column Propagation for new links.** Select this to have runtime column propagation enabled by default when you add a new link in a DataStage job. If you do not select this, you will need to enable it for individual links in the stage editor when you are designing the job.

## **Maintaining Job Log Files**

Every DataStage job has a log file, and every time you run a DataStage job, new entries are added to the log file. To prevent the files from becoming too large, they must be purged from time to time. You can set project-wide defaults for automatically purging job logs, or purge them manually. Default settings are applied to newly created jobs, not existing ones.

To set automatic purging for a project:

- 1 Click the **Projects** tab in the [DataStage Administration window](#) (see page 1-2) to move this page to the front.
- 2 Select the project.
- 3 Click **Properties**. The Project Properties window appears, with the **General** page displayed as shown on [page 1-9](#).
- 4 Select the **Auto-purge of job log** check box.
- 5 Select the Auto-purge action. You can purge jobs over the specified number of days old, or specify the number of jobs you wish to retain in the log. For example, if you specify 10 job runs, entries for the last 10 job runs are kept.
- 6 Click **OK** to set the auto-purge policy. Auto-purging is applied to all new jobs created in the project. You can set auto-purging for existing jobs from the **Clear Log** dialog box. Choose **Job > Clear Log...** from the DataStage Director window to access this dialog box. For more information about purging log file entries, see *"Purging Log File Entries"* in *DataStage Director Guide*.

You can override automatic job log purging for an individual job by choosing **Job > Clear Log...** from the DataStage Director window. For more information, see *"Purging Log File Entries"* in *DataStage Director Guide*.

## Enabling Editing of Internal Reference Information

Select **Enable edition of internal references in jobs** to enable the editing of the **Table definition reference** and **Column definition reference** fields in the column definitions of stage editors. These two fields identify the table definition, and individual columns within that definition, that columns have been loaded from. These fields are enabled on the Stage Editor's **Columns** tab via the **Grid Properties** dialog box. See [Appendix A](#) in *DataStage Designer Guide* for details.

## Protecting a Project

If you are a Production Manager user (see ["Permissions Page"](#) on [page 1-15](#)), you can convert the project to a protected project. This is a special category of project and, normally, nothing can be added, deleted or changed in the project.

**Note** Currently on UNIX systems only root or the administrative user can protect or unprotect a project.

Users can view objects in the project, and perform tasks that affect the way a job runs rather than the jobs design; specifically they can:

- Run jobs

- Set job properties
- Set job parameter default values

Users with Production Manager status can import existing DataStage components into a protected project.

To convert the current project into a protected one, click the **Protect Project** button. A dialog box asks you to confirm the conversion. Click **OK** to proceed. The button changes to an **Unprotect Project** button, which allows you to convert the project back to unprotected status if required.

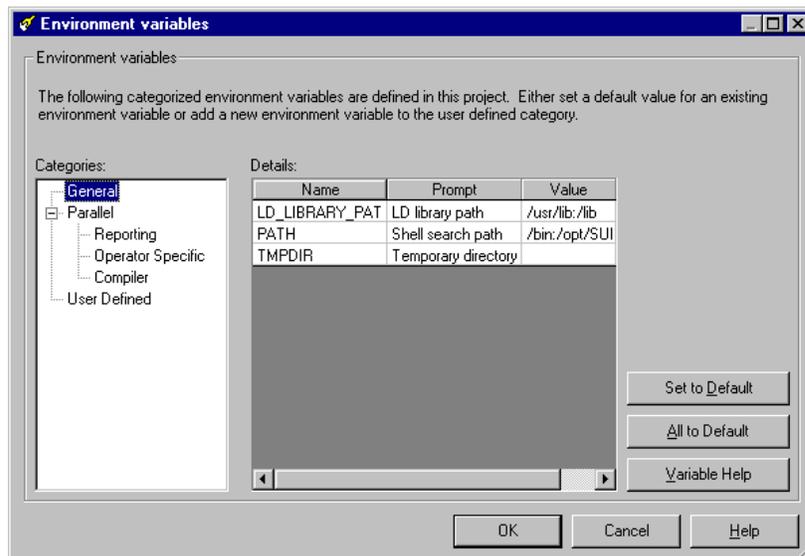
Once a project has been protected, only a Production Manager user (who is *root* or the administrative user on UNIX) can unprotect it.

Protected projects are identified in the Administrator client by the string (Protected) after their name.

## Setting Environment Variables

You can set project-wide defaults for general environment variables or ones specific to parallel jobs from this page. You can also specify new variables. All of these are then available to be used in jobs. They are added to jobs in the same way as job parameters (see *"Specifying Job Parameters"* in *DataStage Designer Guide*).

To define environment variables, click the **Environment...** button. The **Environment Variables** dialog box appears.



Choose the type of environment variable for which you want to set a default from the tree in the left pane. A list of available variables appears in the right pane. You can select a new value for the variable in the **Value** column.

To define a new variable, choose **User Defined**. A dialog box appears asking you to specify the name of the variable and a default value. You can also set a type for a user defined variable. Choose between string (the default) and encrypted. If you choose encrypted, a further dialog box requests you enter the encrypted string value and then confirm it.

Click **Set to Default** to set the selected environment variable to its installed default value.

Click **All to Default** to set the environment variables currently visible to their installed default values.

Click **Variable Help** to get help information about the selected variable.

If you change the setting for an environment variable it will affect all the jobs in the project. If you want to change an environment variable for a particular job, leave the **Value** column empty, and specify the setting of the environment variable via a job parameter (see *"Specifying Job Parameters"* in *DataStage Designer Guide*).

See *"Configuring for Enterprise Edition"* in *DataStage Install and Upgrade Guide* for the environment variables that you need to set for Enterprise Edition. See *"The DataStage Environment on UNIX"* on [page 3-6](#) for a general discussion of environment variables.

## Capturing Process Meta Data

If you have installed the Process MetaBroker, you can capture process meta data from DataStage jobs by selecting the **Generate operational meta data** check box. MetaStage can then collect meta data that describes each job run in this project, and the data warehouse resources it affected. (You can override this setting for individual jobs in the project when you actually run them.)

To facilitate the collection of process meta data, you should create locator tables in any source or target databases that your DataStage jobs access. DataStage uses this information during jobs runs to create a fully qualified locator string identifying table definitions for MetaStage's benefit.

A locator table can be created using the following SQL statement in the database:

```
CREATE TABLE MetaStage_Loc_Info (  
    Computer varchar(64)  
    SoftwareProduct varchar(64)  
    DataStore varchar(64)  
)
```

You should then use an SQL insert statement to populate the table with the relevant information. For example:

```
insert into MetaStage_Loc_Info values ('HAL', 'Oracle8', 'PartsDB')
```

You should also ensure that the locator table is in the current schema, and has select permission. You can include the table in the schema by creating a view called 'MetaStage\_Loc\_Info' that points to the locator table, uses the same column names, and has appropriate permissions.

You can then import table definitions from the data sources and use them in job designs as normal, and the locator information automatically accompanies these (note however, that if you save a table definition to the DataStage Repository, the locator information is not kept with it, you should always load the table definition from the data source). You should then import the table definitions and job designs into MetaStage, using the DataStage MetaBroker. When the jobs run, MetaStage will capture the process meta data, matching the captured table definition to that imported from DataStage.

## Permissions Page

This section describes DataStage user categories and how to change the assignment of these categories to operating system user groups. It also explains how to change the default view of job log entries for DataStage operators.

### DataStage User Categories

To prevent unauthorized access to DataStage projects, you must assign the users on your system to the appropriate DataStage user category. To do this, you must have administrator status (see "[Who Can Administer DataStage?](#)" on page 1-1).

There are four categories of DataStage user:

- DataStage Developer, who has full access to all areas of a DataStage project
- DataStage Production Manager, who has full access to all areas of a DataStage project, and can also create and manipulate protected projects. (Currently on UNIX systems the Production Manager must be root or the administrative user in order to protect or unprotect projects.)
- DataStage Operator, who has permission to run and manage DataStage jobs
- <None>, who does not have permission to log on to DataStage

You cannot assign individual users to these categories. You have to assign the operating system user group to which the user belongs. For example, a user with the user ID *peter* belongs to a user group called

*clerks*. To give DataStage Operator status to user *peter*, you must assign the *clerks* user group to the DataStage Operator category.

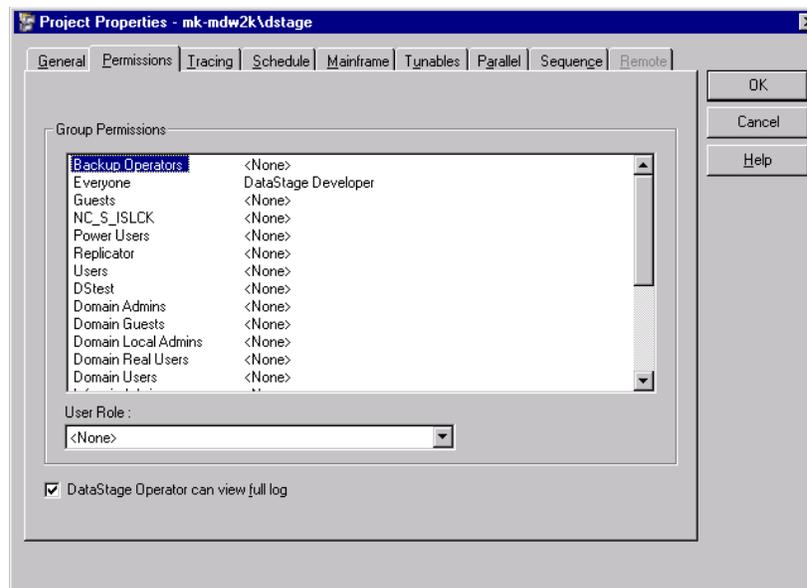
**Note** When you first install DataStage, the Everyone group is assigned to the category DataStage Developer. This group contains all users, meaning that every user has full access to DataStage. When you change the user group assignments, remember that these changes are meaningful only if you also change the category to which the Everyone group is assigned.

You can prevent members of a group from logging on to DataStage by assigning the group to the <None> category.

## Changing User Group Assignments

To change the current user group assignments for a project:

- 1 Click the **Projects** tab in the [DataStage Administration window](#) (see page 1-2) to move this page to the front.
- 2 Select the project.
- 3 Click **Properties** to display the Project Properties window.
- 4 Click the **Permissions** tab to move this page to the front, as shown in the following screen:



**Note** The **Permissions** tab is enabled only if you have administrator status.

The first column in the **Groups** list box shows all the operating system user groups on the server. On UNIX servers, only user

groups listed in */etc/groups* are shown in the list. Domain groups from the primary domain server are preceded by an asterisk (\*).

The second column shows the DataStage user category assigned to the operating system user group.

- 5 To change the DataStage user category for a user group, select the user group in the list box, then choose the category required from the drop-down list.
- 6 Repeat with other groups as required, then click **OK**.

## Changing the Operator's View of Job Log Entries

Entries in a job log file consist of an error message and data associated with the error. By default, the DataStage Director displays both the message and the data when a DataStage operator views the details of an event. To override this setting so that only the error message is visible to an operator, clear the **DataStage Operator can view full log** check box on the **Permissions** page, then click **OK**. Access to the associated data is then restricted to users with Developer rights.

## Limiting Access to Projects

If you are using a UNIX server, you can, if required, control the access users have to particular projects. See "[Limiting Access to Projects](#)" on [page 3-3](#).

## Enabling Tracing on the Server

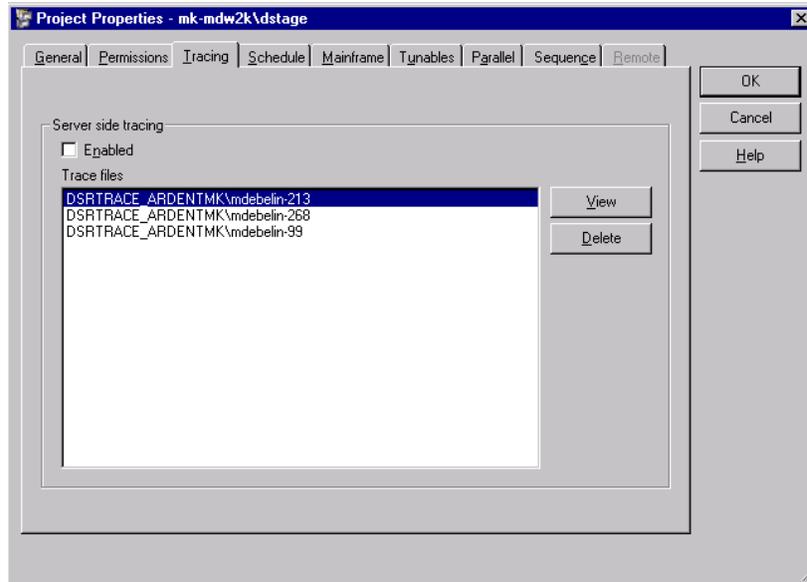
You can trace activity on the server to help diagnose project problems. The default is for server tracing to be disabled. When you enable it, information about server activity is recorded for any clients that subsequently attach to the project. This information is written to trace files, and users with in-depth knowledge of the system software can use it to help identify the cause of a client problem.

If tracing is enabled, users receive a warning message whenever they invoke a DataStage client.

**Note** Server tracing is not compatible with the job administration command **Cleanup Resources**, which you also enable from the Administration client. See "[Enabling Job Administration in the DataStage Director](#)" on [page 1-10](#) for more information.

To enable tracing on the server:

- 1 Click the **Projects** tab in the [DataStage Administration window](#) (see page 1-2) to move this page to the front.
- 2 Select the project.
- 3 Click **Properties** to display the Project Properties window.
- 4 Click the **Tracing** tab to move this page to the front as shown in the following screen:



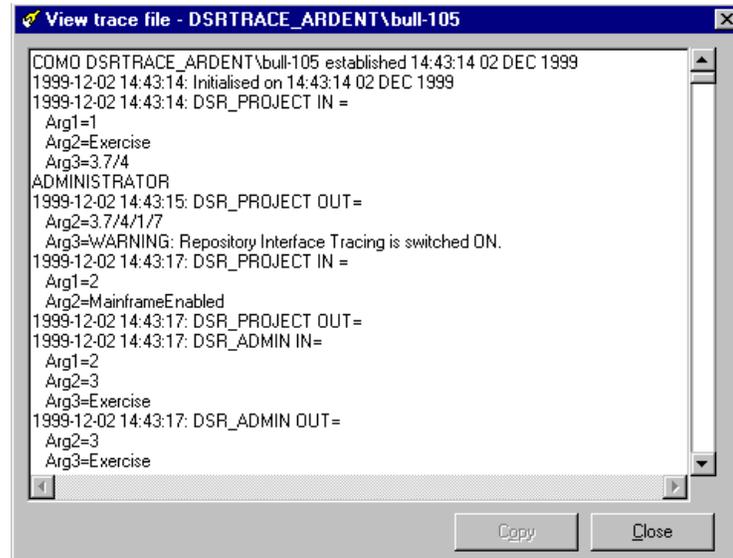
- 5 Click the **Enabled** option button in the **Server side tracing** area.
- 6 Click **OK** to activate server tracing.

Trace files are added to the **Trace files** list box on the **Server** page whenever a client attaches to the project.

To view a trace file:

- 1 Double-click the file name in the **Trace files** list box, or select the file name and then click the **View** button.

The View trace file window displays the trace file:



- 2 To copy trace information to the Clipboard, select text in the View trace file window, then click **Copy**.
- 3 Click **Close** to close the View trace file window.

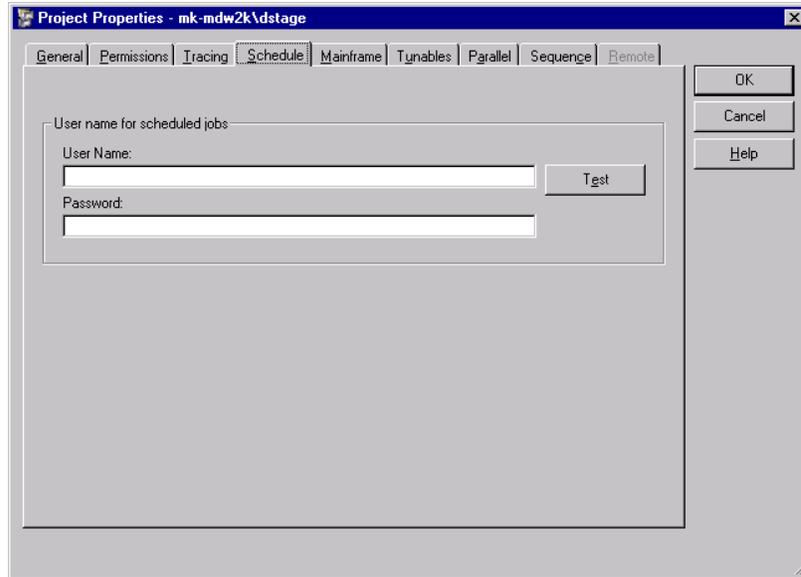
To delete a trace file:

- 1 Select one or more file names in the **Trace files** list box on the **Server** page.
- 2 Click the **Delete** button.

## Schedule Page

*Windows servers only.* DataStage uses the Windows Schedule service to schedule jobs. This means that by default the job runs under the user name of the Schedule service, which defaults to NT system authority. You may find that the NT system authority does not have enough rights to run the job. To overcome this, you can define a user name to run scheduled jobs in a project as follows:

- 1 Click the **Projects** tab in the [DataStage Administration window](#) (see page 1-2) to move this page to the front.
- 2 Select the project.
- 3 Click **Properties** to display the Project Properties window.
- 4 Click the **Schedule** tab to move this page to the front, as shown in the following screen:



- 5 Enter the user name and password you want to use to run the scheduled jobs.
- 6 Click **Test** to test that the user name and password can be used successfully. This involves scheduling and running a command on the server, so the test may take some time to complete.

**Note** The **Test** button is enabled only if you have administrator status (see "[Who Can Administer DataStage?](#)" on page 1-1).

- 7 Click **OK** to save the user name and password.

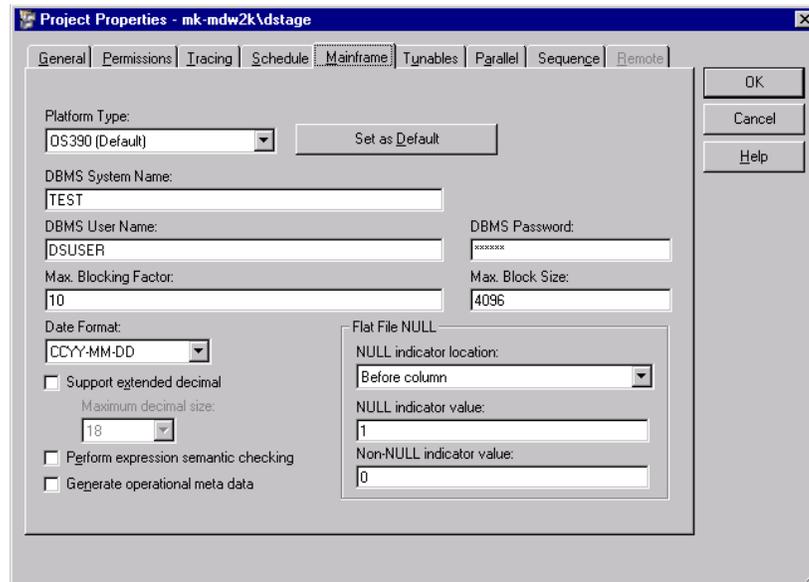
## Mainframe Page

When mainframe jobs are uploaded from the DataStage Designer to a mainframe computer, a JCL script is also uploaded. The script provides information required when the mainframe job is compiled. If you have a project that supports mainframe jobs, you need to define the mainframe job properties for inclusion in the script. You also need to set the default platform type. This platform type is the default when new objects, such as a mainframe routine, are created in the DataStage Manager. The page also allows you to specify details about how nulls are defined in flat files.

To set mainframe job properties and the default platform type:

- 1 Click the **Projects** tab in the [DataStage Administration window](#) (see page 1-2) to move this page to the front.
- 2 Select a project that supports mainframe jobs.
- 3 Click **Properties** to display the Project Properties window.

- Click the **Mainframe** tab to move this page to the front, as shown in the following screen:



**Note** The **Mainframe** tab is enabled only if the selected project supports mainframe jobs.

- Select the default date format for the project from the **Date Format** drop-down list. (The project default can be overridden at job level in the **Job properties** dialog box in the DataStage Designer client.)
- Select the type of platform from the **Platform Type** list.
- Enter the database system name, your user name, and password.
- To change the default maximum block size and maximum blocking factor for the selected platform, enter new values in the **Max. Blocking Factor** and **Max. Block Size** fields.

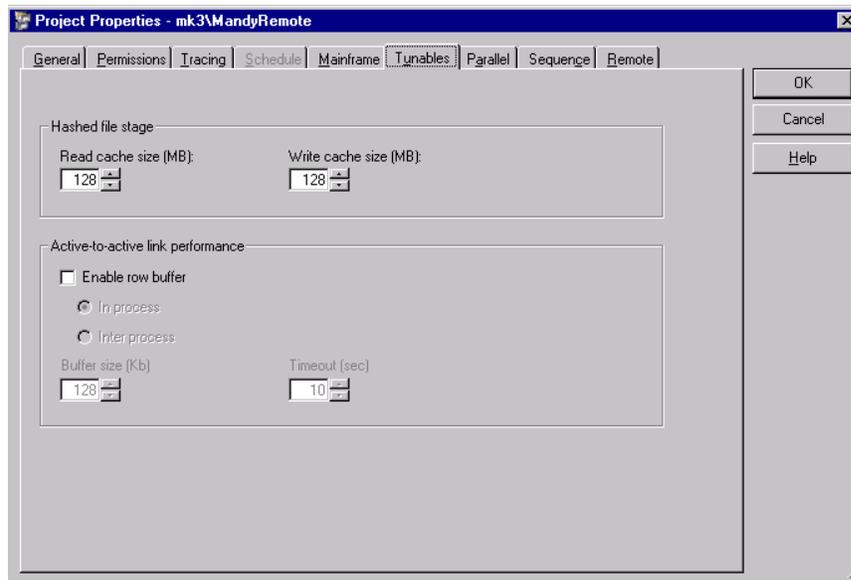
**Note** On an OS/390 platform, the maximum blocking factor can be a value from 1 to 32767. The maximum block size can be zero or a value from 4096 to 32767. If you specify zero, the operating system will determine the optimal block size for you.

- To make the selected platform the default, click the **Set as Default** button. "Default" appears in parentheses after the platform type.
- Select **Perform expression semantic checking** to turn semantic checking on for the expression editor by default in all mainframe jobs in the project. You can override this setting at job level if required.

- 11 Select **Support extended decimal** to enable the use of extended decimal types in the jobs in the project. The **Maximum decimal size field** is then enabled.
- 12 Select **Generate operational meta data** to have mainframe jobs in the project generate process meta data by default which can be used by MetaStage. You can override this setting at job level if required.
- 13 If you have selected **Support extended decimal**, in the **Maximum decimal size** field enter the maximum value that can be specified in the **Length** field of meta data used in mainframe jobs in the project. This can be 18 (the default) or 31.
- 14 From the **Null indicator location** drop-down list, select **Before column** or **After column** to specify the position of NULL indicators in mainframe column definitions.
- 15 In the **Null indicator value** field, specify the character used to indicate nullability of mainframe column definitions. NULL indicators must be single-byte, printable characters. Specify one of the following:
  - A single character value (1 is the default)
  - An ASCII code in the form of a three-digit decimal number from 000 to 255
  - An ASCII code in hexadecimal form of %Hnn or %hnn where 'nn' is a hexadecimal digit (0-9, a-f, A-F)
- 16 In the **Non-Null Indicator Value** field, specify the character used to indicate non-NULL column definitions in mainframe flat files. NULL indicators must be single-byte, printable characters. Specify one of the following:
  - A single character value (0 is the default)
  - An ASCII code in the form of a three-digit decimal number from 000 to 255
  - An ASCII code in hexadecimal form of %Hnn or %hnn where 'nn' is a hexadecimal digit (0-9, a-f, A-F)
- 17 Click **OK** to save your changes.

## Tunables Page

The **Tunables** page allows you to set up caching details for hashed file stages and to set up row buffering to improve performance of server jobs.



### Hashed File Caching

When a Hashed File stage writes records to a hashed file, there is an option for the write to be cached rather than written to the hashed file immediately.

Similarly, when a Hashed File stage is reading a hashed file there is an option to pre-load the file to memory, which makes subsequent access very much faster and is used when the file is providing a reference link to a Transformer stage. (UniData Stages also offer the option of preloading files to memory, in which case the same cache size is used.)

The Hashed file stage area of the Tunables page lets you configure the size of the read and write caches.

- 1 To specify the size of the read cache, enter a value between 0 and 999 in the Read cache size (MB) field. The value, which is in megabytes, defaults to 128.
- 2 To specify the size of the write cache, enter a value between 0 and 999 in the Write cache size (MB) field. The value, which is in megabytes, defaults to 128.
- 3 Click **OK** to save your changes.

## Row Buffering

The use of row buffering can greatly enhance performance in server jobs. Select the **Enable row buffer** check box to enable this feature for the whole project.

There are two types of mutually exclusive row buffering:

- **In process.** You can improve the performance of most DataStage jobs by turning in-process row buffering on and recompiling the job. This allows connected active stages to pass data via buffers rather than row by row.
- **Inter process.** Use this if you are running server jobs on an SMP parallel system. This enables the job to run using a separate process for each active stage, which will run simultaneously on a separate processor.

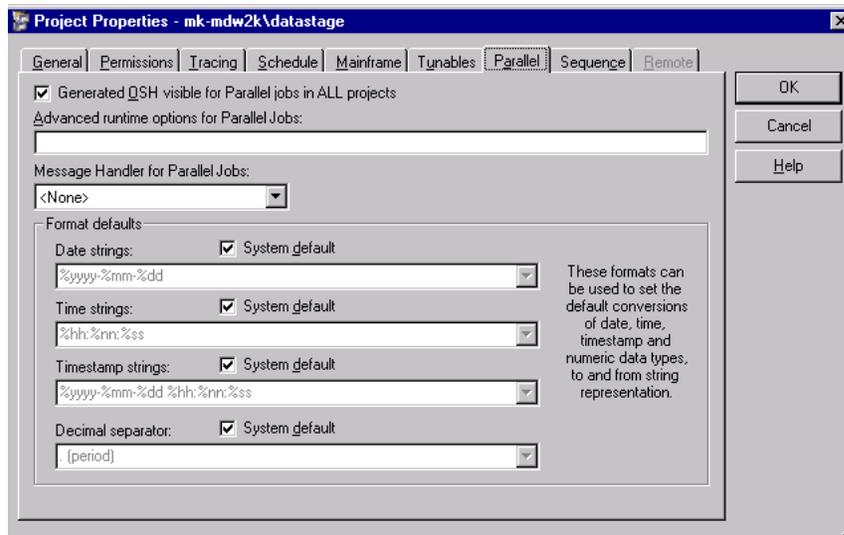
**Note** You cannot use row-buffering of either sort if your job uses COMMON blocks in transform functions to pass data between stages. This is not recommended practice, and it is advisable to redesign your job to use row buffering rather than COMMON blocks.

When you have enabled row buffering, you can specify the following:

- **Buffer size.** Specifies the size of the buffer used by in-process or inter-process row buffering. Defaults to 128 Kb.
- **Timeout.** Only applies when inter-process row buffering is used. Specifies the time one process will wait to communicate with another via the buffer before timing out. Defaults to 10 seconds.

## Parallel Page

The parallel page allows you to specify certain defaults for parallel jobs in the project.



If you select the **Generated OSH visible for Parallel jobs in ALL projects** option, you will be able to view the code that is generated by parallel jobs at various points in the Designer and Director:

- In the **Job Properties** dialog box for parallel jobs.
- In the job run log message.
- When you use the View Data facility in the Designer.
- In the **Table Definition** dialog box.

Note that selecting this option enables this feature for all projects, not just the one currently selected.

The **Advanced runtime options for Parallel Jobs** field allows experienced Orchestrate users to enter parameters that are added to the OSH command line. Under normal circumstances this should be left blank. You can use this field to specify the `-nosortinsertion` and/or `-nopartinsertion` options. These prevent the automatic insertion of sort and/or partition operations where DataStage considers they are required. This applies to all jobs in the project.

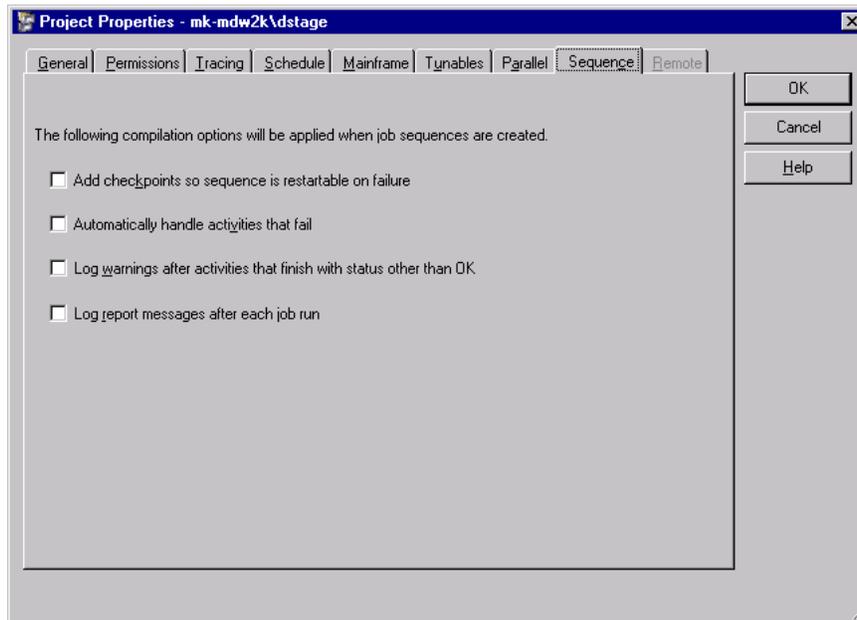
**Message Handler for Parallel Jobs** allows you to specify a message handler for all the parallel jobs in this project. You define message handlers in the DataStage Director. They allow you to specify how certain warning or information messages generated by parallel jobs are handled. Choose one of the pre-defined handlers from the drop-down list.

The Format defaults area allows you to override the system default formats for dates, times, timestamps, and decimal separators. To change a default, clear the corresponding **System default** check box, then either select a new format from the drop down list or type in a new format.

## Sequence Page

Use this page to set compilation defaults for job sequences. You can optionally have DataStage add checkpoints to a job sequence, so that, if part of the sequence fails, you do not necessarily have to start again from the beginning. You can fix the problem and rerun the sequence from the point at which it failed. You can also specify that DataStage automatically handle failing jobs within a sequence (this means that you do not have to have a specific trigger for job failure).

The remaining options allow you to specify that job sequences, by default, log a message in the sequence log if they run a job that finishes with warnings or fatal errors, or a command or routine that finishes with an error status. You can also have the log record a status report for a job immediately the job run finishes..



## Remote Page

This page allows you to specify whether you are:

- Deploying parallel jobs to run on a USS system *OR*
- Deploying parallel jobs to run on a deployment platform (which could, for example, be a system in a grid).

## Deploying on USS Systems

Where you are deploying parallel jobs on a USS system, this page allows you to specify details about:

- The mode of deployment.
- Details of the USS machine being deployed to (this can be used for sending files and remote shell execution).
- Details of a remote shell used to execute commands on the USS machine.
- The location on the USS machine of the deployment files.

Project Properties - mk3\MandyRemote

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

Project settings for Parallel job deployment support:

USS Support:

Deploy standalone Parallel Job scripts

Jobs run under the control of DataStage

USS machine name:

Username:

Password:

Remote shell template:

Base directory name:

Deployed job directory template:

Custom deployment commands:

OK  
Cancel  
Help

For a more detailed description of deploying parallel jobs, see ["Parallel Jobs on USS"](#) in *Parallel Job Developer's Guide*.

The page contains the following fields:

- **Deploy standalone Parallel job scripts.** Select this option to use the standalone method of deployment. This means that parallel jobs on the USS machine are run by you, not by DataStage. If you select only this method, and specify no target machine details, you are also responsible for transferring script files and setting their permissions appropriately.
- **Jobs run under the control of DataStage.** Select this option to run jobs on the USS machine from the DataStage Director. DataStage uses the details you provide in the remainder of this page to FTP the required files to the USS machine and execute it via a remote shell.

You can, if required, have both of the above options selected at the same time. This means that files will be automatically sent and their permissions set, and you can then choose to run them via the DataStage Director, or directly on the USS machine.

The target machine details are specified as follows:

- **Name.** Name of the USS machine to which you are deploying jobs. This must be specified if you have **Jobs run under the control of DataStage** selected. Note that, if you supply this while you have **Deploy standalone Parallel job scripts** only selected, DataStage will attempt to FTP the files to the specified machine. The machine must be accessible from the DataStage server (accessibility from the client is not sufficient).
- **Username.** The username used for transferring files to the USS machine. This can also be used for the remote shell if so specified in the remote shell template.
- **Password.** The password for the username. This can also be used for the remote shell if so specified in the remote shell template.
- **Remote shell template.** Gives details of the remote shell used for setting execution permissions on transferred files and executing deployed jobs if you are running them from the DataStage Designer. The template is given in the form:

```
rshellcommand options tokens
```

For example:

```
rsh -l %u %h %c
```

The tokens allow you to specify that the command takes the current values for certain options. The available tokens are:

- **%h** –host
- **%u** – username
- **%p** – password
- **%c** – command to be executed on remote host

Remote shell details must be supplied if you have **Jobs run under the control of DataStage** selected. If you have **Deploy standalone Parallel job scripts** only selected, DataStage will use any remote shell template you provide to set the required permissions on any transferred job deployment files and perform other housekeeping tasks. You may have security concerns around specifying username and password for remote shell execution in this way. An alternative strategy is to specify a user exit on the USS machine that explicitly identifies permitted users of the remote shell (see ["Allowing User Execution of rsh Without a Password"](#) in *DataStage Install and Upgrade Guide*).

The location for the deployment files on the USS machine are set as follows:

- **Base directory name.** This specifies a base directory on the USS machine. The name of your USS project is added to this to specify a home directory for your project. Each job is located in a separate directory under the home directory. You must specify a full (absolute) pathname, not a relative one).

**Deployed job directory template.** This allows you to optionally specify a different name for the deployment directory for each job. By default the job directory is `RT_SCjobnum` where *jobnum* is the internal jobnumber allocated by DataStage. For example, where you have designated a base directory of `/u/cat1/remote`, and your project is called *USSproj*, you might have a number of job directories as follows:

```
/u/cat1/remote/USSproj/RT_SC101
/u/cat1/remote/USSproj/RT_SC42
/u/cat1/remote/USSproj/RT_SC1958
```

The template allows you to specify a different form of job directory name. The following tokens are provided:

- %j – jobname
- %d – internal number

You can prefix the token with some text if required. For example, if you specified the following template:

```
job_%d
```

The job directories in our example would be:

```
/u/cat1/remote/USSproj/job_101
/u/cat1/remote/USSproj/job_42
/u/cat1/remote/USSproj/job_1958
```

If you choose to use job names for your directory names, note that the following are reserved words, and you must ensure that none of your jobs have such a name:

- buildop
  - wrapped
  - wrapper
- **Custom deployment commands.** This optionally allows you to specify further actions to be carried out after a job in a project marked for standalone deployment has been compiled. These actions normally take place on your DataStage server, but if you have FTP enabled (i.e., have specified FTP connection details in

the target machine area), they take place on the USS machine. In both cases, the working directory is that containing the job deployment files. The following tokens are available:

- %j – jobname
- %d – internal number

You could use this feature to, for example, to tar the files intended for deployment to the USS machine:

```
tar -cvf ../%j.tar *
```

This creates a tar archive of the deployed job with the name *jobname.tar*.

## Deploying on Remote Systems

Where you are deploying parallel jobs on other, deployment-only systems, this page lets you specify details about:

- Provide a location for the deployment.
- Specify names for deployment directories.
- Specify further actions to be carried out at the end of a deployment compile.

For a more detailed description of deploying parallel jobs, see ["Remote Deployment"](#) in *Parallel Job Developer's Guide*.

To specify remote deployment:

- 1 Do **not** select either of the options in the USS support section.
- 2 In the **Base directory name** field, provide a home directory location for deployment; in this directory there will be one directory for each job. This location has to be accessible from the server machine, but does not have to be a disk local to that machine. Providing a location here enables the job deployment features.
- 3 In the **Deployed job directory template** field, optionally specify an alternative name for the deployment directory associated with a particular job. This field is used in conjunction with **Base directory name**. By default, if nothing is specified, the name corresponds to the internal script directory used on the DataStage server project directory, *RT\_SCjobnum*, where *jobnum* is the internal job number allocated to the job. Substitution strings provided are:
  - %j – jobname
  - %d – internal number

The simplest case is just “%j” - use the jobname. A prefix can be used, i.e., “job\_%j”. The default corresponds to RT\_SC%d.

- 4 In the **Custom deployment commands** field, optionally specify further actions to be carried out at the end of a deployment compile. You can specify Unix programs and /or calls to user shell scripts as required.

This field uses the same substitution strings as the directory template. For example:

```
tar -cvf ../%j.tar * ; compress ../%j.tar
```

will create a compressed tar archive of the deployed job, named after the job.

## DataStage License Administration

### Changing Server License Details

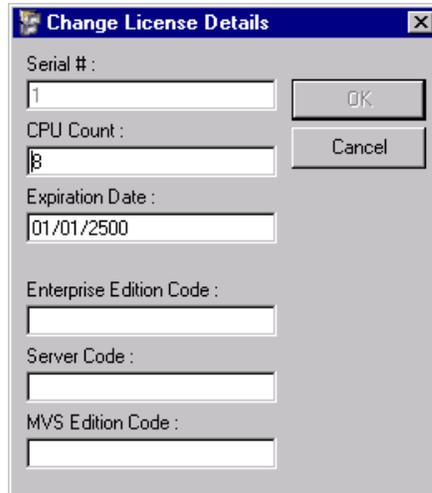
If you have administrator status, you can update the DataStage license from the DataStage Administrator at any time, regardless of whether the current license has already expired.

To relicense DataStage:

- 1 In the DataStage Administration window, click the **License** tab to move the **Licensing** page to the front. The **License Details** area shows the license number, the number of CPUs supported by the license, and the license expiration date.

The **Change...** button is enabled only if you have administrator status (see "[Who Can Administer DataStage?](#)" on page 1-1).

- 2 Click **Change...** . The **Change License Details** dialog box appears:



The screenshot shows a dialog box titled "Change License Details". It contains the following fields and values:

- Serial #: 1
- CPU Count: 8
- Expiration Date: 01/01/2500
- Enterprise Edition Code: (empty)
- Server Code: (empty)
- MVS Edition Code: (empty)

Buttons: OK, Cancel

- 3 Enter the new details exactly as shown on the Ascential license authorization. Note that you have to enter an authorization code for each edition of DataStage you have installed in order to relicense all these editions.
- 4 Click **OK** to relicense the DataStage server.

## Changing Client License Details

You can add details of client license changes which are held on the DataStage server. These can then be used to upgrade particular clients as and when required.

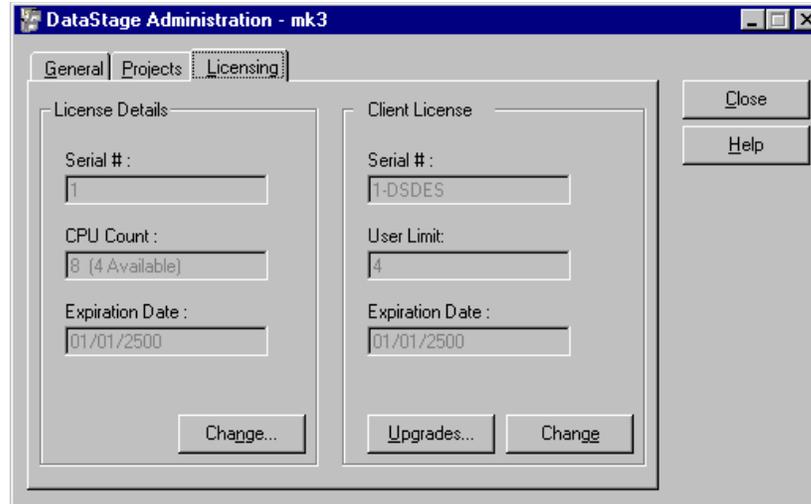
You can upgrade the licenses for a particular client machine manually, by attaching to the server and either entering the new details or asking the server to select a suitable upgrade from its list.

You can also instruct the server to automatically upgrade client licenses as and when client machines attach.

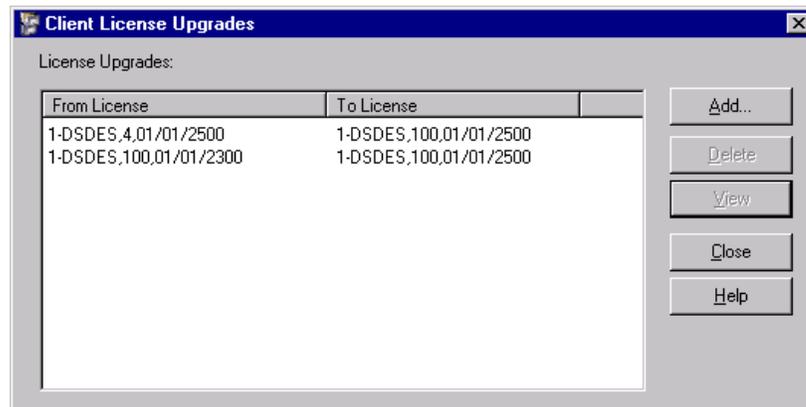
## Adding to the Upgrade List

To add upgrades to the server's list:

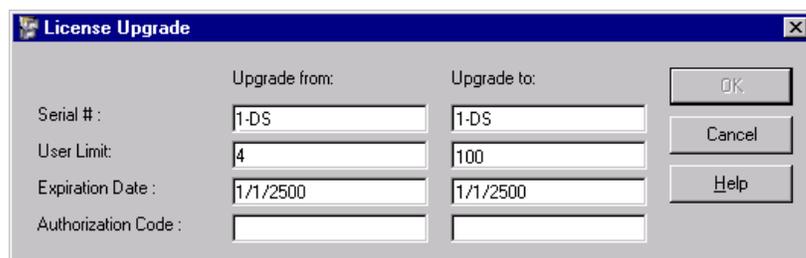
- 1 In the DataStage Administration window, click the **License** tab to move the **Licensing** page to the front. The **Client license area** shows the details of the current client license for the client you are connected from.



- 2 Click the **Upgrades...** button. The **Client License Upgrades** dialog box appears:



- 3 Click the **Add...** button to add a new upgrade to the list. The **License Upgrade** dialog box appears:



- 4 In the **Upgrade from** area, enter the details for a current client license. This must be valid, except for the expiration date, which can be expired.

- 5 In the **Upgrade to** area, enter the details for the new client license. When you are done, click **OK**. The new upgrade appears in the **Client License Upgrades** dialog box.

To delete an existing upgrade:

- 1 Select an upgrade in the **Client License Upgrades** dialog box.
- 2 Click the **Delete** button. The client license upgrade is deleted from the server's list and is no longer available for immediate or automatic upgrade.

To view and update an existing upgrade:

- 1 Select an upgrade in the **Client License Upgrades** dialog box.
- 2 Click the **View** button. The **License Upgrade** dialog box appears, displaying the details for the selected upgrade. You must enter a new authorization code for the **Upgrade from** area, then you can enter totally new details for the **Upgrade to** area. When you are done, click **OK**.

## Upgrading Client Licenses using the Upgrade List

To have DataStage upgrade client licenses automatically when a client connects:

- 1 In the DataStage Administration window, click the **License** tab to move the **Licensing** page to the front.
- 2 In the **Client License** area, click the **Change...** button. The **Change Client License Details** dialog box appears:



- 3 Ensure that the **Automatic License Upgrade** option is selected (it is selected by default). Whenever a client machine attaches to the server, it will search its list for suitable upgrades to apply to the client.

To upgrade manually (and immediately) from the upgrade list:

- 1 In the DataStage Administration window, click the **License** tab to move the **Licensing** page to the front.
- 2 In the **Client License** area, click the **Change...** button. The **Change Client License Details** dialog box appears:
- 3 Ensure that the **Automatic License Upgrade** option is NOT selected (it is selected by default). The **Upgrade** button is enabled.
- 4 Click the **Upgrade** button, the server searches for suitable upgrades to apply to your client, and will inform you as to its success or failure.

## Upgrading Client Licenses Manually

To upgrade the license for the client machine from which you are connected, without using the upgrade list facility:

- 1 In the DataStage Administration window, click the **License** tab to move the **Licensing** page to the front.
- 2 In the **Client License** area, click the **Change...** button. The **Change Client License Details** dialog box appears.
- 3 Enter the details for the license to which you are upgrading.
- 4 Click **OK**.

## Issuing DataStage Engine Commands

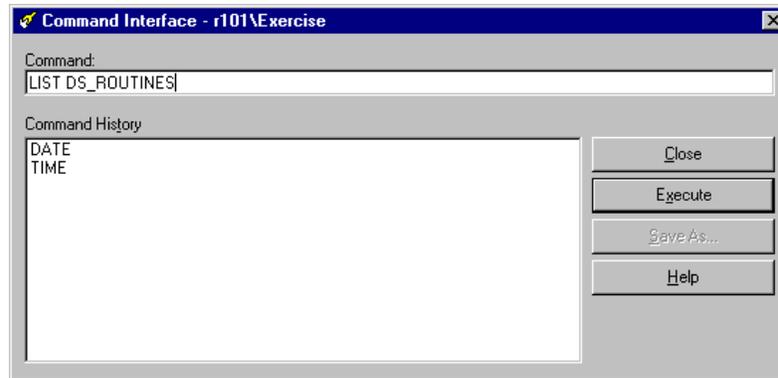
The Administration client lets you issue DataStage Engine commands directly from a selected project rather than having to use a Telnet session.

**Note** You cannot issue the following DataStage Engine commands from the client: LOGOUT, LO, QUIT, Q, and OFF.

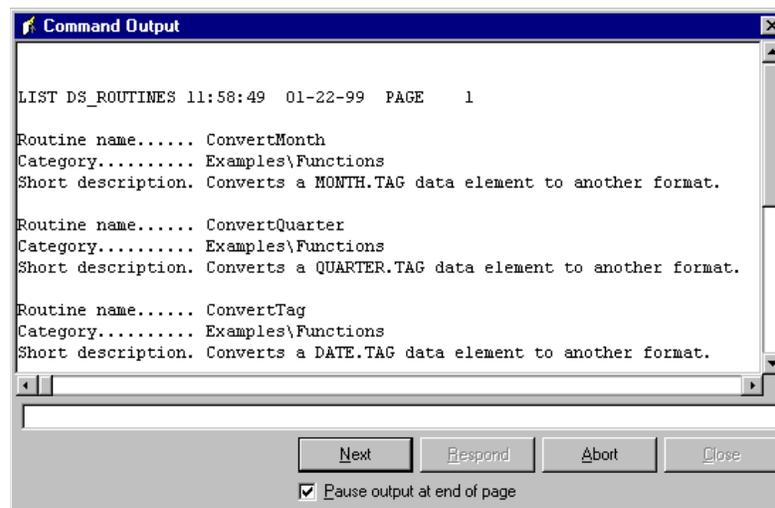
**Warning** You should take care when issuing DataStage Engine commands that manipulate the NLS settings of your system. Incorrect use of these commands could disrupt both DataStage and the rest of your system.

To issue a DataStage Engine command from the Administration client:

- 1 Click the **Projects** tab in the [DataStage Administration window](#) (see page 1-2) to move this page to the front.
- 2 Select the project.
- 3 Click **Command**. The **Command Interface** dialog box appears:



- 4 Either type the DataStage Engine command you want to execute into the **Command** field, or double-click a command in the **Command History** list box. A command selected from the command history appears in the **Command** field, where you can edit it.
- 5 Click **Execute**. The command is added to the **Command History** list box and the Command Output window appears, displaying the results of the command:



The command output pauses at the end of the first page. Click **Next** to scroll to the next page of output. If you want to switch paging off, clear the **Pause output at end of page** check box at the bottom of the **Command Output** dialog box.

- 6 If the command requires further input, the Command Output window displays a prompt. Enter the response in the field below the command output display, then click **Respond**.
- 7 When the command has run to completion, click **Close**. This window closes, and the **Command Interface** dialog box reappears.

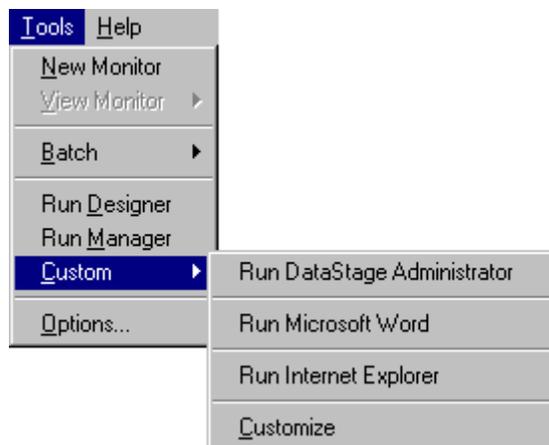
- 8 If you want to save commands to the VOC file on the server, select one or more commands from the **Command History** list box, then click **Save As...** . A single command is saved as a sentence and multiple commands as a paragraph. Enter the name of the sentence or paragraph in the **Save As** dialog box, then click **OK**.

**Note** You cannot save a sentence or paragraph that contains a call to itself.

- 9 Click **Close** to close the **Command Interface** dialog box.

## Customizing the Tools Menu

The DataStage Designer, Manager, and Director windows each have a **Tools** menu that is used to call other DataStage components. You can customize the **Tools** menus to call other software directly from DataStage. In particular, you can customize it so you can run the DataStage Administrator from any of these clients:



The **Tools** menu is customized for each user, and is saved as part of the user's DataStage environment. As well as adding and removing entries in the menu, you can specify the order in which they appear, and separate the entries with dividers. You can add up to 20 entries to the menu, including dividers.

### Adding a Tool

To add an entry to a **Tools** menu:

- 1 From the menu bar, choose **Tools** ► **Custom** ► **Customize**. The **Customize** dialog box appears.
- 2 Click **Add...** . The **Add Tool** dialog box appears.

- 3 Enter the path of the program you want to add, or click ... to browse for the path. If the program you entered cannot be found, you are prompted to change the path. Click **No** to add the entry anyway if you intend to install the program later.
- 4 Click **OK**. The program's run file details appear in the **Customize** dialog box.
- 5 Edit the **Menu Text** field as required. (This field contains the text that appears in the **Tools** menu.)
- 6 Edit the **Arguments** field to specify any arguments required by the program. Click > for a list of variables you can use in this field to specify, for example, the current project, user name, or host.
- 7 Optionally add text in the **Status bar** field. This text appears in the status bar when you call the program from the **Tools** menu of the DataStage Designer.
- 8 Click **OK**. The entry is added to the **Tools** menu.

## Changing the Tools Menu Order

To change the order of entries on the **Tools** menu:

- 1 From the menu bar, choose **Tools > Custom > Customize**. The **Customize** dialog box appears.
- 2 Select the entry you want to move in the **Menu contents** field.
- 3 Click the **Up** or **Down** arrow to move the entry to the required position.
- 4 Click **OK** to save your new settings and close the **Customize** dialog box.

**Note** You can only change the order of the **Custom** area on the **Tools** menu.

## Adding a Separator

To add separators between entries on the **Tools** menu:

- 1 From the menu bar, choose **Tools > Custom > Customize**. The **Customize** dialog box appears.
- 2 Click **Add...**. The **Add Tool** dialog box appears.
- 3 Select the **Add as menu separator** check box and click **OK**. The separator appears in the **Menu contents** field.
- 4 Click the **Up** or **Down** arrow to move the separator to the required position.
- 5 Click **OK** to save your new settings and close the **Customize** dialog box.

## DataStage NLS

If DataStage NLS (National Language Support) is enabled, DataStage will support the language you specified during the install without any further configuration. But if your requirements change, you can reconfigure NLS to support different languages. For more information, see [Chapter 4](#), “NLS Configuration.”



# 2

## Windows Server Administration

This chapter describes the administration tasks that need to be carried out when the DataStage server runs under Windows. The administration tasks include:

- Starting the Windows services used by DataStage
- Accessing ODBC databases and UniVerse systems
- Installing DataStage packages (including plug-ins)
- Installing MetaBrokers

You must be logged on as a member of the Windows Administrators group to perform most server administration tasks.

### User Roles on Windows Systems

There are four categories of DataStage user:

- DataStage Developer, who has full access to all areas of a DataStage project
- DataStage Production Manager, who has full access to all areas of a DataStage project, and can also create and manipulate protected projects.
- DataStage Operator, who has permission to run and manage DataStage jobs
- <None>, who does not have permission to log on to DataStage

The default user roles for DataStage are that all members of the Windows Administrators group have the DataStage Production Manager role, and that all members of the Windows Everyone group have the DataStage Developer user role. We recommend that you set

up three user groups using the Windows user administration tools: one for operators, one for developers, and one for production managers. You can then populate these groups with users and assign the appropriate DataStage user role to each one as described in ["DataStage User Categories"](#) on [page 1-15](#). When you have done this, you should then set the user role for the Windows Everyone group to <None>.

**Warning** We do not recommend that DataStage Production Manager or Developer roles are assigned to the Windows Users group, as this group includes NT AUTHORITY\INTERACTIVE and can therefore allow privileges to unauthorized users such as the built-in 'guest' user.

## Starting the Schedule Service

The DataStage Director uses the Windows Schedule service to schedule jobs, and if you try to schedule jobs when this service is not running, an error message appears. Start the Schedule service as follows:

- 1 Log on to the DataStage server. Choose **Start > Settings > Control Panel**. The Control Panel window appears.
- 2 Choose **Services**. The **Services** dialog box appears.
- 3 Scroll down and choose **Schedule**.
- 4 Click **Start**. The service is started.
- 5 If Startup is not listed as Automatic, click **Startup...**, then choose **Automatic**. This means the Schedule service restarts automatically when the system reboots.
- 6 Click **Close** to close the **Services** dialog box.

**Note** If you encounter any problems running scheduled jobs, see ["Windows Scheduling Problems"](#) on [page 5-2](#).

## Stopping and Restarting the Server Engine

From time to time, you may need to stop or restart the DataStage server engine manually, for example, when you want to shut down the physical server.

To stop the server engine:

- 1 Close all DataStage clients.
- 2 Log on to the DataStage server. Choose **Start > Settings > Control Panel > DataStage**. The **DataStage Control Panel** appears.
- 3 Click **Stop all services**. The services are stopped.

To restart the server engine:

- 4 Log on to the DataStage server. Choose **Start > Settings > Control Panel > DataStage**. The **DataStage Control Panel** appears.
- 5 Click **Start all services**. The services are started.

Alternatively, you can start and stop services from the generic **Services** dialog box. To stop the server engine in this way:

- 1 Close all DataStage clients.
- 2 Log on to the DataStage server. Choose **Start > Settings > Control Panel > Services**. The **Services** dialog box appears.
- 3 Select **DataStage Engine Resource Service**, then click **Stop**. The **Stopping** dialog box notifies you that **DataStage Telnet Service** will also be stopped. Click **OK**.
- 4 Select **DSRPC Service**, then click **Stop**.

To restart server engine:

- 1 Log on to the DataStage server. Choose **Start > Settings > Control Panel > Services**. The **Services** dialog box appears.
- 2 Select **DSRPC Service**, then click **Start**.
- 3 If Startup is not listed as Automatic, click **Startup...**, then click **Automatic**. This ensures the service restarts automatically whenever the server is rebooted.
- 4 Repeat steps 2 and 3 for **DataStage Engine Resource Service**, and **DataStage Telnet Service**.

## Accessing ODBC Databases

If you want to access an ODBC database from DataStage, you must first install ODBC drivers for the ODBC data source on the DataStage server. You can install your own preferred drivers.

**Note** Do not use trial versions of ODBC drivers on an active DataStage system. When the trial version expires, these drivers tend to display reminder screens that need user input. The DataStage server cannot handle these screens and the job or meta data import hangs.

When you have installed a suitable ODBC driver:

- Add the path of the directory containing the driver DLL to the Path system environment variable.
- Configure the ODBC data source as a System Data Source, not a User Data Source.

Otherwise DataStage cannot connect to the ODBC data source.

## Accessing UniVerse Systems

To access files in UniVerse systems from DataStage, you must define the systems in the *uvodbc.config* file in the relevant DataStage project directory on the server. The *uvodbc.config* file is a text file containing comments that describe how to add additional entries. The default *uvodbc.config* file looks like this:

```
***
*** To get to any ODBC source other than UniVerse, you need entries
*** that look as follows (the data source must also be configured
*** via the operating system's own mechanisms):
***
*** <data source name>
*** DBMSTYPE = ODBC
***
*** NOTE THAT DATASTAGE VERSION 2.0 ONWARDS DOES NOT NEED THE ABOVE
*** ENTRIES ON WINDOWS NT SYSTEMS, IT READS THE NT REGISTRY DIRECTLY
***
*** The local DataStage Server Engine is available via the data
*** source name "localuv" as defined below - please do not alter
*** this entry!
***
*** To access a remote UniVerse database, you need another entry
*** similar to that for localuv but with a remote host name in
*** place of "localhost".
***
*** To access a (coresident) UniVerse on the local machine, you
*** need to specify your local machine name or IP address in place
*** of "localhost".
***
*** Note that the spaces around the " = " signs are required, and
*** the data source name must be enclosed in angle brackets "<>".
***
[ODBC DATA SOURCES]
<localuv>
DBMSTYPE = UNIVERSE
network = TCP/IP
service = uvserver
host = localhost
```

You must not edit or delete the default DataStage entry in the file. To add an entry for a remote UniVerse system, position your cursor at

the bottom of the file and add the following lines, with the italicized variables changed to suit your system:

```
<name>  
DBMSTYPE = UNIVERSE  
network = TCP/IP  
service = uvserver  
host = hostname:portnumber
```

*name* identifies the remote UniVerse system in any way you find useful and must be enclosed in angle brackets, as shown.

*hostname* specifies the host where UniVerse is installed and must be recognized as a node name by your TCP/IP system.

*Portnumber* specifies the port number for the connection. This is 31438 by default on UniVerse systems.

To access a co-resident UniVerse on the local machine, specify your local machine name or IP address in place of *hostname*.

Save the file when you have added all the host names you require.

## Installing DataStage Packages

Your DataStage supplier may provide packaged jobs and components to add to DataStage. DataStage plug-ins are provided as packages. Plug-ins extend the value of DataStage by providing additional functionality, or extending the reach of DataStage to external systems.

To install a package into DataStage:

- 1 Either load the media containing the package, or copy the package files or directories into a temporary directory on your hard disk, as described in the instructions supplied with the package.
- 2 In the Ascential DataStage program folder, click the **DataStage Package Installer** icon to start the installation program.
- 3 Follow the instructions on the screen to install the package into DataStage.
- 4 Delete the files from the temporary directory, if necessary.

Most plug-ins are available on the DataStage installation CD, and the plug-in directory (Packages) on the CD contains documentation for these plug-ins. Between major DataStage releases, new plug-ins often become available, and you can download and install them from the Ascential Software TechLink Web site. The following Web site gives access to TechLink:

**<http://www.ascentialsoftware.com/service/techinfo.htm>**

To use TechLink, you need a product serial number and a valid support contract.

The DataStage Package Installer also accepts command line options, allowing you to call it as part of another installation program.

## Package Installer Command Line Options

To run the Package Installer from the command line, use the following command:

```
%DSPackagePath%\setup.exe package=dirpath
    project=proj_name | allprojects
    [logfile=pathname] [appendlog]
```

**DSPackagePath** is a preset environment variable that specifies the location of the Package Installer setup program. The setup program takes the following options:

Option	Description
package= <i>dirpath</i>	<b>package</b> is mandatory. <i>dirpath</i> is the directory in which the package is located. If the specified path does not exist or does not point to a directory, an entry is added to the log file (see the <b>logfile</b> option description) and setup exits.
project= <i>proj_name</i>	Installs the package into the project named <i>proj_name</i> . You can specify only a single project. The command line must contain either <b>project</b> or <b>allprojects</b> . If it includes both, <b>allprojects</b> overrides <b>project</b> .
allprojects	Installs the package into all the projects on the current server. This option overrides <b>project</b> . You must include either <b>project</b> or <b>allprojects</b> on the command line.
logfile= <i>pathname</i>	Identifies the file that logs information about the progress of the package installation. The pathname is optional. If you do not specify a directory or the specified directory does not exist, it defaults to <i>dshome</i> . If you do not include <b>logfile</b> on the command line, the default is to use the file <i>install.log</i> in the directory <i>dshome</i> .
appendlog	Specifies that information should be appended to the log file, rather than overwrite the existing information.

For example, the command:

```
%DSPackagePath%\setup.exe package=c:\ikt\package1 allprojects appendlog
```

installs the package in directory *c:\ikt\package1* into all the projects on the current server. The default log file will be used (*dshome\install.log*) and new log information appended to the file. (*dshome* is the DataStage server directory, which is normally *C:\Program Files\Ascential\DataStage\ServerEngine*.)

## Installing MetaBrokers

To use MetaBrokers in DataStage, you need to install the MetaBrokers for each data warehousing tool with which you want to exchange meta data.

MetaBrokers are provided on a separate CD, which can be obtained from your DataStage supplier.

To install one or more MetaBrokers:

- 1** Load the MetaBrokers CD.
- 2** Run the top-level **setup.exe** program on the CD.
- 3** The MetaBroker installation wizard appears and guides you through the installation process. This includes supplying license information for each MetaBroker you select.

Documentation is supplied for each MetaBroker in its respective directory on the MetaBrokers CD.

**Warning** If you previously had MetaBrokers installed with DataStage 3.5, you must uninstall these before installing the latest DataStage MetaBrokers (otherwise the install will fail).



## UNIX Server Administration

This chapter briefly describes the administration tasks that need to be carried out when the DataStage server runs under UNIX. These tasks include:

- Checking installation requirements
- Setting up user roles
- Tuning UNIX for DataStage
- Configuring the server daemon environment
- Starting and stopping the server engine
- Accessing UniVerse systems
- Accessing ODBC databases
- Installing DataStage packages (including plug-ins)
- Observing DataStage UNIX server restrictions
- Accessing remote files via NFS

You must be logged in as *root* or *dsadm* to perform most server administration tasks.

More detailed information is given in "[Installing on UNIX Systems](#)" and "[Upgrading UNIX Systems](#)" in *DataStage Install and Upgrade Guide*. It is also a good idea to look at the Read Me file supplied with your release, as this contains late breaking news.

The term *dshome* is used throughout this chapter to refer to the DataStage installation directory.

## Installation Requirements

The DataStage UNIX server is installed from the DataStage CD. Before installing, you should consider the following points:

- If you are installing on a system that has never had DataStage installed, then you may need to alter some of your kernel parameters. See ["Tuning UNIX for DataStage"](#) on [page 3-4](#).
- If you are upgrading DataStage from a release earlier than 7.5, you need to set your shared library search path to include the *lib* subdirectory of the DataStage server home directory *dshome* (specified in the *.dshome* file). See ["Environment Requirements for dsrpcd"](#) on [page 3-5](#).
- If you require access to ODBC data sources, you will need to install appropriate ODBC driver software. See ["Accessing ODBC Databases"](#) on [page 3-9](#).
- If you intend to install any plug-in additions to DataStage, you may need to set other environment variables. See ["Environment Requirements for dsrpcd"](#) on [page 3-5](#).

To install the DataStage server, run the *install.sh* program from the CD.

## User Roles on UNIX Systems

There is a designated administrative user for DataStage. By default this is *dsadm*, but you can specify a different administrative user when installing (see ["Types of Install"](#) in *DataStage Install and Upgrade Guide*).

All users who are going to use DataStage need to be in the same primary group as the DataStage administrative user.

There are four categories of DataStage user:

- DataStage Developer, who has full access to all areas of a DataStage project
- DataStage Production Manager, who has full access to all areas of a DataStage project, and can also create and manipulate protected projects.
- DataStage Operator, who has permission to run and manage DataStage jobs.
- <None>, who does not have permission to log on to DataStage

By default the DataStage administrative user and *root* have the Production Manager role, and everyone else in the same primary

group has the DataStage Developer role. You can change this arrangement by assigning secondary groups to users. **The primary group must stay the same as for the administrative user**, but you can set up three secondary groups for Production Manager, Developer, and Operator roles. You can then assign the DataStage roles to these secondary groups using the Administrator as described in "[DataStage User Categories](#)" on [page 1-15](#). When you have done this, set the user role for the primary group to <None>.

## Limiting Access to Projects

You can, if required, control the access users have to particular projects; this requires one UNIX group per project in addition to the single UNIX group to which the software itself is assigned. Consider the following scenario as an example:

- 1 The ACME Company has a UNIX server with DataStage installed on it. They want to have a test project and a production project on the same machine. They do not want the test users to be able to edit the jobs in production, and they do not want the production users to be able to edit the jobs in the test project.
- 2 When they installed DataStage, they created the *dsadm* user with the primary group of 'dstage'. During the install they created a project called 'prod' and second project called 'test'. After the install, all the files in both projects are owned by the *dsadm* user, and the group is *dstage*.
- 3 *root* now must perform the following tasks:
  - a Create two new UNIX groups, one called 'dstest' and one called 'dsprod'.
  - b Make the *dsadm* user a member of BOTH of these new groups.
  - c Make all the test users a member of both the *dstage* group and the *dstest* group.
  - d Make the production users a member of the *dstage* group and the *dsprod* group.
- 4 *Dsadm* should now change the group of all the files in the test project to 'dstest' and all the files in the production project to 'dsprod'.
- 5 Finally, *dsadm* should go to each project directory and set its SGID bit. This forces files created in that directory to have the same group id as the directory in which they were created. This can be done as follows:

```
chmod g+s dirpath
```

where *dirpath* is the absolute path of the DataStage project directory.

You may like to tighten up security further by removing the access to all DataStage file of all users who are not in the primary DataStage group. To do this:

- As *root*, edit the *ds.rc* file in the *DSEngine/sample* directory, change the *umask* setting to *007*. After making this change, stop the DataStage engine and restart it.

## Tuning UNIX for DataStage

The DataStage Server runs jobs and other activities as UNIX processes. The demands the DataStage Server makes on the operating system depend on:

- The particular DataStage activity being performed
- The number of DataStage processes active on the system

You should check that the tunable kernel parameters meet the minimum requirements specified in "[Set Kernel Parameters](#)" in *DataStage Install and Upgrade Guide*.

If necessary, rebuild the kernel with changes to these parameters. For more information on rebuilding the kernel, see the UNIX manuals supplied with your system.

## Running Out of File Units

You may need to perform some additional tuning if you come across the problem of running out of file units. The DataStage server uses the parameter *MFILES* and the kernel parameter *NOFILES* to determine the number of open files allowed. The number of open files allowed is *NOFILES* - *MFILES*. If you encounter problems and run out of file units, you can decrease the value of *MFILES* in the DataStage server file *uvconfig* or increase the value of *NOFILES* for your operating system.

The *uvconfig* file resides in the *dshome* directory, for example, */u1/dsadm/Ascential/DataStage/DSEngine/uvconfig*.

If you change the value of *MFILES*, you need to stop and restart the DataStage server "[Stopping and Restarting the Server Engine](#)" on [page 3-6](#).

## The DataStage Server Daemon (*dsrpcd*)

The daemon for the server engine is called *dsrpcd*. It must be running to allow connection of DataStage clients.

By default it is started when the server is installed, and should start whenever you restart your machine. If you cannot connect from a DataStage client, check that *dsrpcd* is running.

If you need to start *dsrpcd* manually, stop and restart the DataStage engine manually as described in "[Stopping and Restarting the Server Engine](#)" on [page 3-6](#).

If you cannot start the *dsrpcd* process, there may be an open connection from a DataStage client. This can happen when you shut down the DataStage server without closing all client connections, thus causing the port to be unavailable for use with new connections. To avoid an open connection, ensure that all client connections are closed (that is, no active sockets) before shutting down the DataStage server. Enter the following command at the UNIX prompt:

```
# netstat | grep dsrpcd
```

This command produces no output when all *dsrpcd* connections are closed. At this point, you can terminate the *dsrpcd* process by shutting down the DataStage server. Without shutting down the client processes, sockets are released on most platforms after the timeout period of about 6 - 10 minutes, depending on system tuning.

### Environment Requirements for *dsrpcd*

The *dsrpcd* daemon is the means by which processes that represent DataStage jobs are started. The environment that DataStage processes inherit when they are started is the same environment as for *dsrpcd*. ODBC drivers and some plug-ins require that certain directories are included in the shared library environment variable setting for *dsrpcd*. The shared library environment path is set in *dshome/dsenv*. the name of the environment variable depends on the platform:

Platform	Environment Variable
Solaris	LD_LIBRARY_PATH
HP-UX	SHLIB_PATH
AIX	LIBPATH
Compaq Tru64	LD_LIBRARY_PATH
Linux	LD_LIBRARY_PATH

Some databases also require specific environment variables (for example, Oracle requires ORACLE\_HOME and Sybase requires SYBASE). To access these databases from DataStage, *dsrpcd* must have these environment variables set to appropriate values.

For details of exact environment variable requirements, see "[The DataStage Environment on UNIX](#)" on [page 3-6](#) and the sections on Configuring Plug-ins and Configuring ODBC access in "[Installing on UNIX Systems](#)" and "[Upgrading UNIX Systems](#)" in *DataStage Install and Upgrade Guide*.

## Stopping and Restarting the Server Engine

From time to time you may need to stop or restart the DataStage server engine manually, for example, when you wish to shut down the physical server.

A script called *uv* is provided for these purposes.

To stop the server engine, use:

```
# dshome/bin/uv -admin -stop
```

This shuts down the server engine and frees any resources held by server engine processes.

To restart the server engine, use:

```
# dshome/bin/uv -admin -start
```

This ensures that all the server engine processes are started correctly.

You should leave some time between stopping and restarting. A minimum of 30 seconds is recommended.

## The DataStage Environment on UNIX

Various environment variables can control how DataStage jobs behave on a UNIX system. (Parallel jobs in particular make extensive use of environment variables to control the way they behave.) This section describes how DataStage handles environment variables and describes administrator actions that can affect them.

DataStage derives the environment in which to run a job as follows:

- 1 All server process, job processes, and client connection helper processes are ultimately child processes of *dsrpcd* and inherit its environment settings. The environment for *dsrpcd* is derived from

the */etc/profile* and *dshome/dsenv* scripts. (Note that client connections DO NOT pick up any per-user environment settings from their *\$HOME/.profile* script).

- 2 Environment variable settings for particular projects can be set in the DataStage Administrator, see "[Setting Environment Variables](#)" on [page 1-13](#). Any settings for a specific environment variable will override any settings inherited from dsrpcd as described above.
- 3 Environment variables that have been set up for the project using the DataStage Administrator can be added to a job's parameter list and their values set at run time via the **Job Run Options** dialog box (see "[The Job Run Options Dialog Box](#)" in *DataStage Designer Guide*).

There are also some more specialized ways in which job environments can be controlled:

- If you require to run DataStage jobs from within a shell script or from the UNIX command line you can ensure that it inherits the current user's environment by using dsjob with the `-local` argument (see "[DataStage Development Kit \(Job Control Interfaces\)](#)" in the *Server Job Developer's Guide*).
- If a job finds an environment variable `$ENVVARNAME` (i.e., the name of an existing environment variable prefixed by a dollar) and its value is set to `'$ENV'`, then DataStage will determine the current value of `ENVVARNAME` and use its value for `$ENVVARNAME`. This value remains set until explicitly changed for the job.
- If a job finds an environment variable `$ENVVARNAME` and its value is set to `'$PROJDEF'`, then DataStage will determine the current value of `ENVVARNAME` and use its value for `$ENVVARNAME`. If the value of the environment variable is subsequently changed, the job will pick up the new value without the need for recompiling.
- If a job finds an environment variable `$ENVVARNAME` and its value is set to `'$UNSET'`, then DataStage will explicitly unset that environment variable.

## Accessing UniVerse Systems

To access files in UniVerse systems from DataStage, you must define the systems in the *uvodbc.config* file in the relevant DataStage project directory on the server. The *uvodbc.config* file is a text file containing comments that describe how to add additional entries. The default *uvodbc.config* file looks like this:

```

*** To get to any ODBC source other than UniVerse, you need entries
*** that look as follows (the data source must also be configured
*** via the operating system's own mechanisms):
***
*** <data source name>
*** DBMSTYPE = ODBC
***
*** The local DataStage Server Engine is available via the data
*** source name "localuv" as defined below - please do not alter
*** this entry!
***
*** To access a remote UniVerse database, you need another entry
*** similar to that for localuv but with a remote host name in
*** place of "localhost".
***
*** To access a (coresident) UniVerse on the local machine, you
*** need to specify your local machine name or IP address in place
*** of "localhost".
***
*** Note that the spaces around the " = " signs are required, and
*** the data source name must be enclosed in angle brackets "<>".
***
[ODBC DATA SOURCES]
<localuv>
DBMSTYPE = UNIVERSE
network = TCP/IP
service = uvserver
host = localhost

```

You must not edit or delete the default DataStage entry in the file. To add an entry for a remote UniVerse system, position your cursor at the bottom of the file and add the following lines, with the italicized variables changed to suit your system:

```

<name>
DBMSTYPE = UNIVERSE
network = TCP/IP
service = uvserver
host = hostname:portnumber

```

- *name* identifies the remote UniVerse system in any way you find useful and must be enclosed in angle brackets, as shown.
- *hostname* specifies the host where UniVerse is installed and must be recognized as a node name by your TCP/IP system.
- *Portnumber* specifies the port number for the connection. This is 31438 by default on UniVerse systems.

To access a co-resident UniVerse on the local machine, specify your local machine name or IP address in place of *hostname*.

Save the file when you have added all the host names you require.

## Accessing ODBC Databases

Ascential provides an OEM version of the Data Direct ODBC driver pack with this release. These files are copied to your system during all install and upgrade installations of DataStage. They are installed in *dshome/branded\_odbc*.

Information about configuring DataStage to use these drivers, and about using alternative ODBC drivers, is given in the Section "Configuring ODBC Drivers" in *"Installing on UNIX Systems"* and *"Upgrading UNIX Systems"* in *DataStage Install and Upgrade Guide*. *"Troubleshooting"* in that guide contains information about troubleshooting faulty ODBC connections.

**Note** You should not use trial versions of ODBC drivers. The DataStage server cannot handle the screens highlighting the evaluation status of the driver and may hang.

### Adding Data Sources to *uvodbc.config*

Once you have configured an ODBC connection, you must add an entry to the *uvodbc.config* file for every ODBC data source that you want to access from DataStage.

The *uvodbc.config* file is normally in the DataStage server directory (*dshome*), but you can also have different *uvodbc.config* files in each project directory to configure different data sources. This is useful where you configure a data source that is known to some projects but not others. By default, DataStage searches the current project directory for a *uvodbc.config* file and, if it finds one, uses this in preference to a file in *dshome*.

Add an entry to *uvodbc.config* in the form:

```
<name>
```

```
DBMSTYPE = ODBC
```

*name* is the ODBC data source name, enclosed in angle brackets as shown. Also note that the spaces either side of the equal sign are required.

See ["Accessing UniVerse Systems"](#) on [page 3-7](#) for more information about editing the *uvodbc.config* file. For more information about configuring ODBC connections, see ["ODBC Connection Problems on UNIX"](#) on [page 5-5](#).

## Installing DataStage Packages

Your DataStage supplier may provide packaged jobs, components, or plug-in stage types to add to DataStage. Plug-ins extend the value of DataStage by providing additional functionality, or extending the reach of DataStage to external systems.

To install the packages on the DataStage server:

- 1 Either load the media containing the package, or copy the package files or directories into a temporary directory on your hard disk.
- 2 Log on to the DataStage server as *root* or *dsadm*.
- 3 At the UNIX prompt, enter the command:

```
# dshome/bin/dspackinst
```

*dshome* is the installation directory of the DataStage server engine and is specified in the file */.dshome*.

- 4 Follow the instructions on the screen to install the package into DataStage.
- 5 Delete the files from the temporary directory, if necessary.

Most plug-ins are available on the DataStage UNIX installation CD, and the plug-in directory (*Packages*) on the CD contains documentation for these plug-ins. Between major DataStage releases, new plug-ins often become available, and you can download and install them from the Ascential Software TechLink Web site. The following Web site gives access to TechLink:

**<http://www.ascentialsoftware.com/service/techinfo.htm>**

To use TechLink, you need a product serial number and a valid support contract.

## Server Restrictions

### Connecting to the Server

The **Omit** option on the DataStage client **Attach to Project** dialog box is applicable only for systems with Windows servers. It should not be used when connecting to UNIX servers, and you should make all your DataStage users aware of this.

## Scheduling Jobs

When using the DataStage Director to schedule a job to run on a UNIX server, the **Every** option is available only with a single day of the week or month (it is not possible to use it with multiple days within a week or month).

The DataStage server uses a combination of the UNIX *at* and *cron* system commands to provide flexible job scheduling. A scheduled job runs under the user name of the connected Director client session that scheduled the job. You must ensure that any DataStage operators who need to schedule jobs have adequate permissions to invoke the *at* and *cron* commands. See your UNIX documentation for details of how to do this.

The *cron* command does not set environment variables for the jobs that it runs. Therefore some DataStage jobs, particularly those whose components use third-party software, may not run. If this is the case, you may need to manually edit the *crontab* entry to set the required environment variables.

## Accessing Remote Files via NFS

You can configure your DataStage server to allow access to remote files across the Network File System (NFS).

To do so, you must reconfigure the ALLOWNFS configurable parameter in the *uvconfig* file in the UV account directory *dshome*:

- 1 Edit *uvconfig* to change ALLOWNFS from 0 to 1.
- 2 Execute *uvregen* in the *bin* directory of the UV account directory (*/dshome/uv/bin/uvregen*).
- 3 Stop and restart the server engine, see "[Stopping and Restarting the Server Engine](#)" on page 3-6.

When using NFS, you will have to use fully qualified UNIX pathnames in the format *machine/path* rather than the Windows UNC-type pathnames.

**Warning** Using NFS for remote file access in this way may not use the locking mechanisms of the database or file system on the remote machine. Consequently, you should exercise extreme caution when using this method to avoid data integrity violations.



# 4

## NLS Configuration

DataStage has built-in National Language Support (NLS). This means DataStage can:

- Process data in a wide range of languages
- Use local formats for dates, times, and money
- Sort data according to local rules

Using NLS, DataStage holds data in Unicode format. This is an international standard character set that contains many of the characters used in languages around the world. DataStage maps data to or from Unicode format as required.

Each DataStage project has a map and a locale assigned to it during installation. The map defines the character set that the project can use. The locale defines the local formats for dates, times, sorting order, and so on (sorting order only for parallel jobs), that the project should use. The DataStage client and server components also have maps assigned to them during installation to ensure that data is transferred in the correct format.

DataStage different mechanisms for implementing NLS for server and parallel jobs, and so you set map and locale details separately for the two types of job. Under normal circumstances, the two settings will match.

From the DataStage Administration window, you can check which maps and locales were assigned during installation and change them as required.

## Changing Project Maps

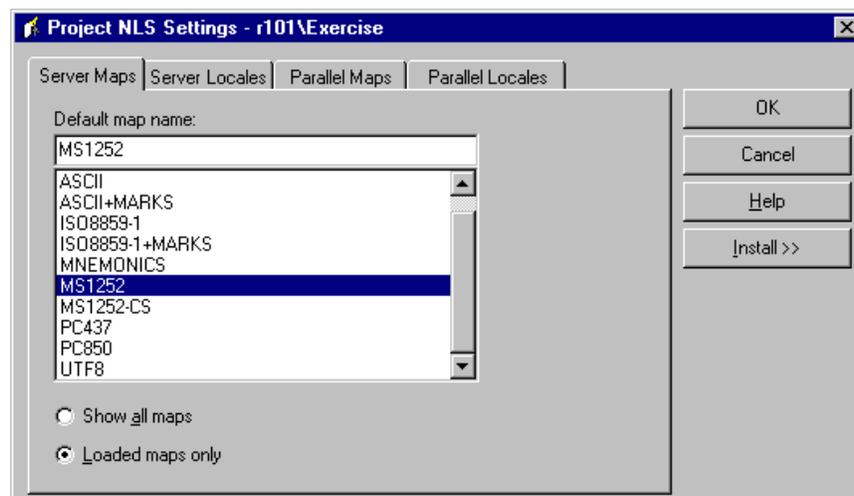
To view or change a project map:

- 1 Click the **Projects** tab in the DataStage Administration window to move this page to the front.
- 2 Select the project.
- 3 Click **NLS...** . The Project NLS Settings window appears.

**Note** If the **NLS...** button is not active, you do not have NLS installed. You must install NLS from the DataStage Master Setup screen as described in *DataStage Install and Upgrade Guide* before you can access the windows described in the rest of this chapter.

- 4 Choose whether you want to set the project map for server jobs or parallel jobs and choose the **Server Maps** or **Parallel Maps** tab accordingly.

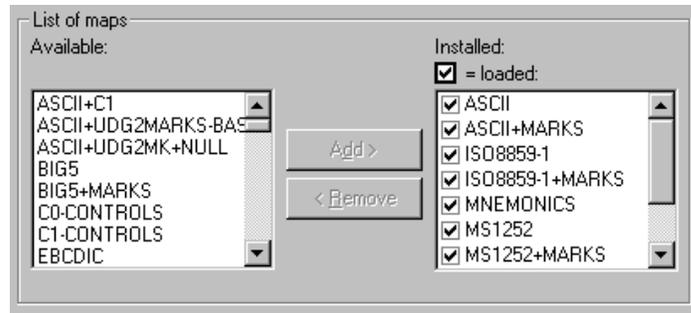
### Server Job Project Maps



The **Default map name** field shows the current map that is used for server jobs in the project. By default, the list shows only the maps that are loaded and ready to use in DataStage. You can examine the complete list of maps that are supplied with DataStage by clicking **Show all maps**.

To change the default map name for the project, click the map name you want to use, then click **OK**.

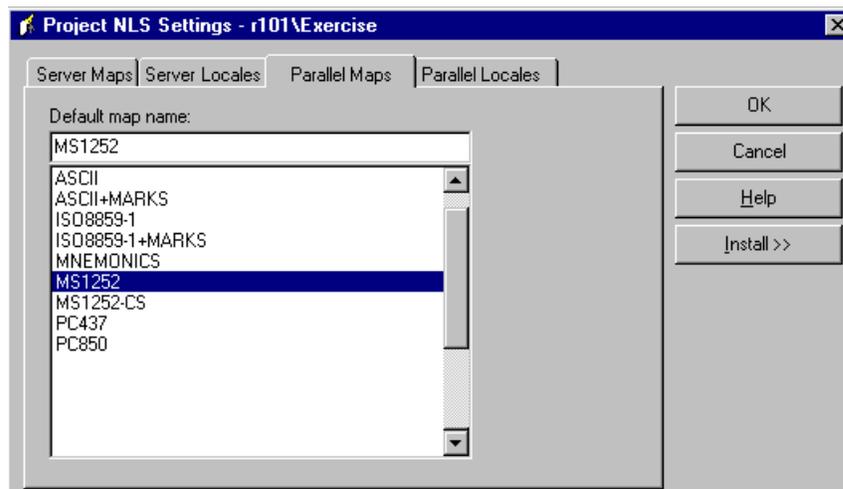
To install a map into DataStage, click **Install>>** to see additional options on the **Maps** page:



The **Available** list shows all the character set maps that are supplied with DataStage. The **Installed/loaded** list shows the maps that are currently installed. To install a map, select it from the **Available** list and click **Add >**. The map is loaded into DataStage ready for use the next time the server is restarted. If you want to use the map immediately, you can restart the server engine. For more information, see ["Stopping and Restarting the Server Engine"](#) on page 2-2 for Windows servers or ["Stopping and Restarting the Server Engine"](#) on page 3-6 for UNIX servers.

To remove an installed map, select it from the **Installed/loaded** list and click **< Remove**. The map is unloaded the next time the server is rebooted or the server engine is restarted.

## Parallel Job Project maps



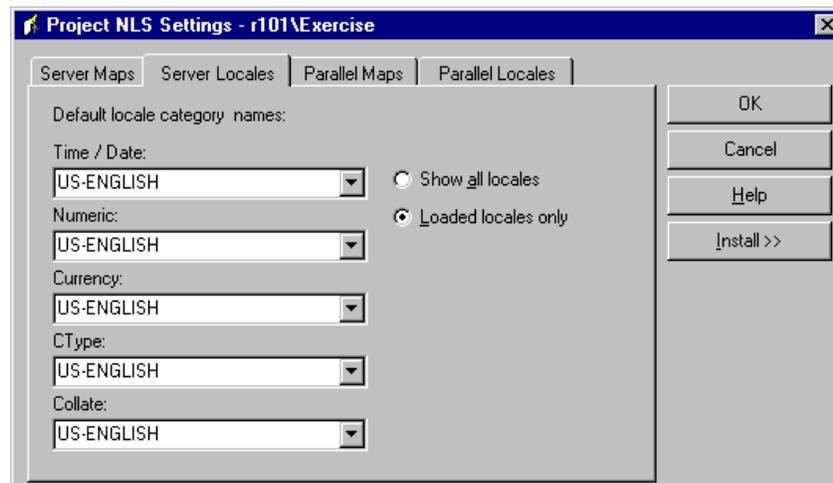
The **Default map name** field shows the current map that is used for parallel jobs in the project. The list shows only the maps that are loaded and ready to use in DataStage. Double-click the map you want to make the default map.

**Note** If you are connected to a Windows server, rather than a UNIX server, the maps shown are provisional. If you subsequently move the project to a UNIX server, where parallel stages can be compiled and run, DataStage will attempt to locate the equivalent of the chosen map on the UNIX system.

## Changing Project Locales

To view or change default project locales, having opened the Project NLS Settings Window, click the **Server Locales** tab or **Parallel Locales** tab as appropriate.

### Server Job Locales



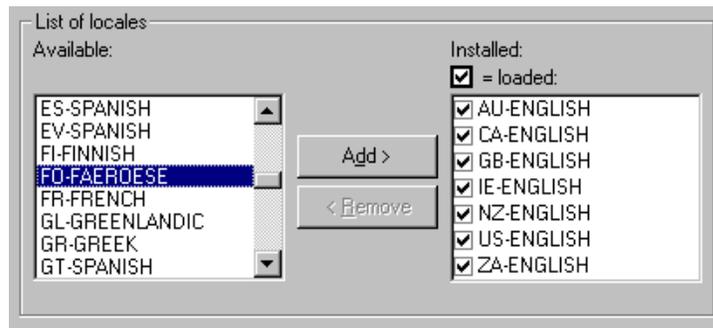
This page shows fields for the default project locales in five categories:

- **Time/Date** – The format for dates and times, for example, 31 Dec 1999 or 12/31/99 are two ways of expressing the same date that may be used in different locales.
- **Numeric** – The format used for numbers, including the thousands separator and radix (decimal) delimiter.
- **Currency** – The format for monetary strings, including the type and position of the currency sign (\$, £, F, DM, and so on).
- **CType** – The format for character types. This includes defining which characters can be uppercase or lowercase characters in a language.
- **Collate** – The sort order used for a language.

By default, each field has a drop-down list of the locales that are loaded and ready to use. To change a locale in any category, select the locale you want from the drop-down list. Click **OK** when you have completed your changes. You can examine the complete list of locales that are supplied with DataStage by clicking **Show all locales**, then clicking a category drop-down list. These locales must be installed and loaded into DataStage before you can use them.

## Installing and Loading Locales

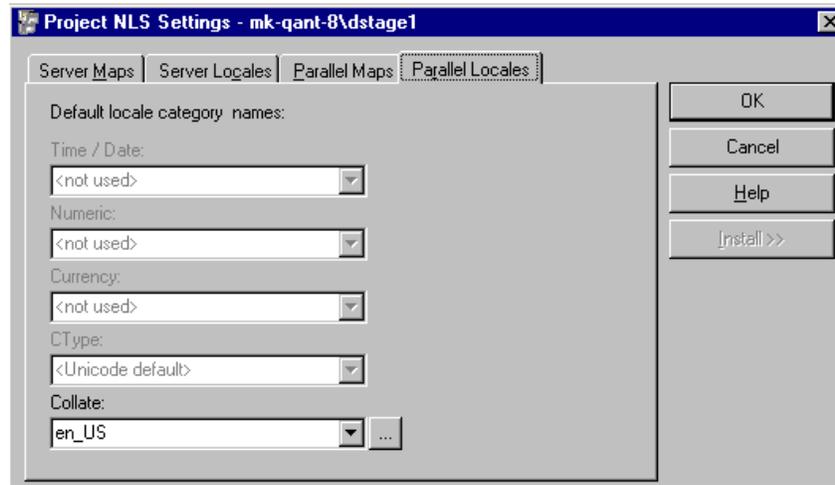
To install a locale, click **Install>>** to display further options on the **Locales** page:



The **Available** list shows all the locales that are supplied with DataStage. The **Installed/loaded** list shows the locales that are currently installed. To install a locale, select it from the **Available** list and click **Add>**. The locale is loaded into DataStage ready for use the next time the server is restarted. If you want to use the locale immediately, you can restart the server engine. For more information, see ["Stopping and Restarting the Server Engine"](#) on [page 2-2](#) for Windows servers or ["Stopping and Restarting the Server Engine"](#) on [page 3-6](#) for UNIX servers.

To remove an installed locale, select it from the **Installed/loaded** list and click **<Remove**. The locale is unloaded the next time the server is rebooted or the server engine is restarted.

## Parallel Job Locales



Only the collate category is used for parallel jobs. Choose a locale from the drop down list of installed locales.

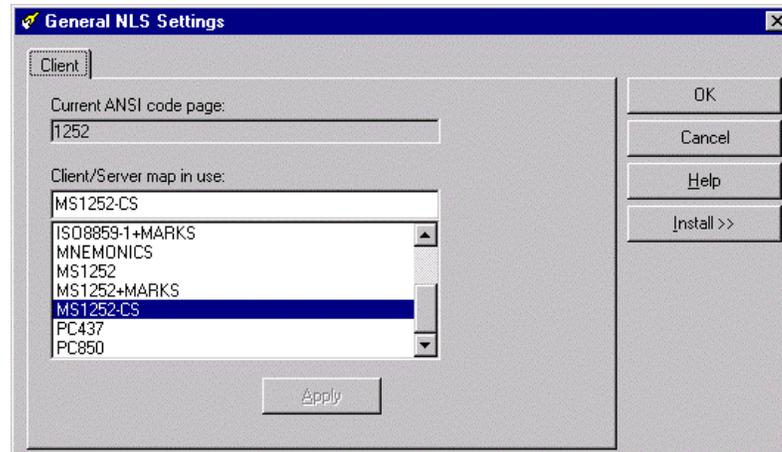
The Browse button allows you to browse for text files that define other collation sequences.

## Client/Server Maps

When you installed the DataStage server, you specified the language that you want DataStage to support. DataStage automatically sets the language supported on the DataStage clients to match what you specified for the server. But if you access the DataStage server from a different client, data may not be mapped correctly between the client and the server.

To prevent this from happening, you must change the client maps. To view the current mapping:

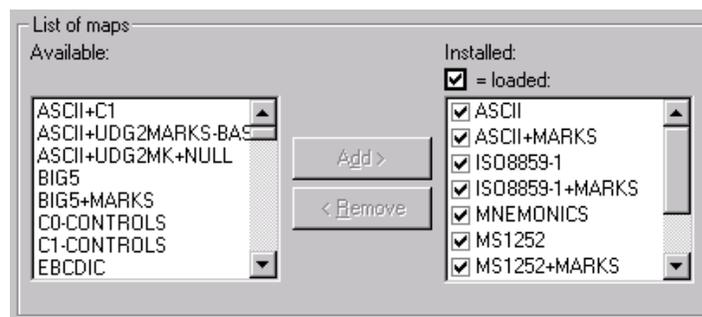
- 1 Click the **General** tab on the DataStage Administration window to move this page to the front.
- 2 Click the **NLS...** button. The General NLS Settings window appears:



The **Current ANSI code page** field is informational only, and contains the current Microsoft code page of the client. The code page is independent of the current project or server. The **Client/Server map in use** field shows the name of the map being used on the server for the current client session. The list shows all loaded maps.

If you select a map and click **Apply**, DataStage attempts to set this map for all clients connecting to the current server that use the code page shown. The mapping is tested, and may be rejected if it is not appropriate.

To install further maps into DataStage, click **Install>>** to display further options on the **Client** page:



DataStage uses special maps for client/server communication, with names ending in "-CS" (for Client Server). You should always choose one of these maps for this purpose.

The **Available** list shows all the character set maps that are supplied with DataStage. The **Installed/loaded** list shows the maps that are currently installed. To install a map, select it from the **Available** list and click **Add>**. The map is loaded into DataStage ready for use at the next time the server is restarted. If you want to use the map immediately, you can restart the server engine. For more information, see ["Stopping and Restarting the Server Engine"](#) on [page 2-2](#) for

Windows servers or "[Stopping and Restarting the Server Engine](#)" on [page 3-6](#) for UNIX servers.

To remove an installed map, select it from the **Installed/loaded** list and click **<Remove**. The map is unloaded the next time the server is rebooted or the server engine is restarted.

# 5

## Troubleshooting

This chapter gives some ideas for troubleshooting problems in DataStage, including:

- Error messages
- Problems with scheduled jobs
- ODBC connection problems on UNIX
- DataStage client to UNIX server connections

### Error Messages

The first indication of a problem may be an error message. If you see an error message in DataStage, it is important to remember that there are potentially several sources for the error:

- The DataStage client
- The DataStage server
- The network connection between the client and the server
- The DataStage server engine

Some message boxes include a **Copy** button that allows you to copy the error message to the Clipboard if you want to save the information that it contains. This is always advisable if you need to contact Ascential technical support about the problem.

## Scheduled Jobs

A DataStage user can schedule jobs to run at convenient times when the system is less busy. (This task is carried out from the Job Schedule view in the DataStage Director window.) DataStage does not have its own separate scheduling program. Instead, whenever a DataStage user schedules a job, the control of that job is handed over to the underlying operating system on the server. This means that if scheduled jobs do not run correctly, the problem usually lies with the operating system configuration on the DataStage server.

The following sections offer some advice on troubleshooting the scheduler on both Windows and UNIX servers.

### Windows Scheduling Problems

On Windows servers, job scheduling is carried out by the Schedule service. If your scheduled job did not run, there are a number of things you can do to identify the cause.

#### Check That the Schedule Service Is Running

First check whether the Schedule service is running, and if it is not, start it. See ["Starting the Schedule Service"](#) on [page 2-2](#).

#### Look at the Schedule Log

If the scheduled job still does not run, examine the schedule log for more clues. The schedule log is a text file called `dsr_sched.log` in the project directory on the server. This file records any problems that occurred before control was handed over from the scheduler to DataStage. (After that point, messages are written to the appropriate job log file.) This file would contain a message if, for example, the server password you specified has expired.

#### Test the User Name and Password

If you specified a user name and password to run all scheduled jobs in a project, make sure that you test the user name and password. This process is described in ["Schedule Page"](#) on [page 1-19](#).

- If the test works correctly but scheduled jobs still do not run, check that the user name you specified has permission to read and write to the project directory.
- If the test fails, there may be a problem with the user rights for the user name you specified. In which case, try the procedure in the next section.

## Check the User Rights

If the Windows Schedule service on the DataStage server is not running under the default user name, try this procedure to ensure that the Schedule service has the correct user rights:

- 1 From the Windows server, choose **Start > Programs > Administrative Tools > User Manager**.
- 2 Choose **Policies > User Rights...** . The **User Rights Policy** dialog box appears.
- 3 Select the **Show Advanced User Rights** check box.
- 4 From the **Right** list, select **Act as part of the operating system**.
- 5 If the user name for the renamed Scheduled service is not in the **Grant To** list:
  - a Click **Add...** . The **Add Users and Groups** dialog box appears.
  - b Click **Show Users** to add user names to the **Names** list.
  - c Select the user name required and click **Add**.
  - d Click **OK** to close the **Add Users and Groups** dialog box.
- 6 From the **Right** list, select **Replace a process level token**. Repeat the procedure in step 5, then continue at step 7.
- 7 From the **Right** list, select **Increase quotas**. Repeat the procedure in step 5, then continue at step 8.
- 8 Click **OK** to save the changes.

## Problems Using the Scheduler on Non-English Language Systems

If you are running DataStage on a system with a language other than English, and encounter problems when scheduling jobs to run on specific days of the week, you can try localizing the days of the week for each project. (The AT command, which performs the Windows scheduling, only accepts day names in the local language.)

To localize the days names:

- 1 Go to the project directory for your first project. This is located on the DataStage server, by default in the folder `Ascential\DataStage\Projects`.
- 2 Edit the file `DSPParams` (use a text editor such as Notepad for this).
- 3 Add the localized days of the week to the end of the file. The following is an example of what you might add for a French system:

```
[SCHEDULER]
MONDAY=L
TUESDAY=M
WEDNESDAY=ME
THURSDAY=J
FRIDAY=V
SATURDAY=S
SUNDAY=D
```

You may have to experiment with which day names the local AT command will accept. If in doubt, enter the full name (for example, LUNDI, MARDI, etc).

- 4 Repeat the process for each of your projects.

You may find that you get an error message equivalent to 'There are no entries in the list' when you use the scheduler on a non-English language system. This is output by the AT command and passed on by the Director. You can prevent the Director passing on the message as follows:

- 1 Identify a unique part of the message that the AT command is outputting (for example, 'est vide' in French).
- 2 For each project, add the following line to its DSParams file:

```
NO ENTRIES=est vide
```

The AT command usually accepts other keywords besides days of the week in English. If your system does not, you can add localized versions of the additional keywords NEXT, EVERY and DELETE to your projects as follows:

- 1 Edit the DSparams file for each project.
- 2 Add a line of the form:

```
KEYWORD=localized_keyword
```

For each keyword. For example:

```
NEXT=Proxima
```

## UNIX Scheduling Problems

On UNIX servers, the scheduling of DataStage jobs is handled by the *at* and *cron* commands. These offer slightly different options than the Windows Schedule service, and the problems associated with them also vary.

### Viewing Scheduled Jobs

On UNIX servers, jobs are scheduled for each user. That means a user can only view jobs that are scheduled for the current user name. For a DataStage administrator, this means it is not possible to get a quick

overall view of all the DataStage jobs that are scheduled to run over a particular period. The only way to find out which jobs are scheduled is to examine the files in the *cron* directory for each user ID. The naming and location of these files varies from system to system. For more information, see the reference page for the *cron* command.

### If a Job Does Not Run

If a scheduled job does not run, check that the user who scheduled the job has permission to use the *cron* command. To do this, examine the *cron.allow* and *cron.deny* files which contain lists of users who can and cannot use the command. The location of these files varies from system to system. For more information, see the reference page for the *cron* command.

## Job Termination Problems

If you experience delays in the termination of a DataStage job when it is run, empty the `&PH&` directory. There is a `&PH&` directory in each DataStage project directory, which contains information about active stages that is used for diagnostic purposes. The `&PH&` directory is added to every time a job is run, and needs periodic cleaning out.

To clear the file from within DataStage:

- 1 Ensure there are no DataStage jobs running anywhere on the system.
- 2 From the DataStage Administrator, go to the **Projects** page, select the project whose file you want to clear and click the **Command** button. The **Command Interface** dialog box opens.
- 3 Type the following into the command field:  

```
CLEAR.FILE &PH&
```
- 4 Click **Execute** to run the command and clear the file.

## ODBC Connection Problems on UNIX

DataStage relies on third-party ODBC drivers to connect to ODBC data sources. This means that it is not always easy to track down problems in an ODBC connection. The following sections contain some ideas for troubleshooting ODBC connection problems.

## Check the Shared Library Environment

If you see a message similar to this one:

```
ld.so.1: uvsh: fatal: libxxxx: can't open file: errno=2
```

check that the ODBC driver's shared library has been added to the environment variable used to locate shared libraries (see "[Accessing ODBC Databases](#)" on [page 3-9](#)).

## Check Symbolic Links

If you have moved shared libraries to a new directory or have installed a new ODBC driver manager, you may have broken symbolic links used by the server engine.

To reset the symbolic links to a new directory, run this command at the UNIX prompt:

```
# dshome/bin/dspackinst relink.uvlibs pathname
```

*dshome* is the home directory of the DataStage server engine.

*pathname* is the full pathname of the directory containing the shared libraries.

To reset links for a new ODBC driver manager:

- 1 Install the ODBC driver manager according to the vendor's instructions.
- 2 Determine where the ODBC shared library *libodbc.xx* resides. For example, the library for the Intersolv driver lives in *\$ODBCHOME/dlls*, and the library for the Visigenics driver lives in *\$ODBCHOME/libs*.
- 3 Exit any DataStage client applications.
- 4 Run the *relink.uvlibs* command as described above.
- 5 Restart any DataStage client applications.

## DataStage Client to UNIX Server Connections

If you cannot connect from a DataStage client to a UNIX server, check that the *dsrpcd* daemon is running. The *dsrpcd* daemon is started when the DataStage server is installed, and should start automatically when you reboot. If the daemon has stopped for some reason, restart it with the following command:

```
dshome/bin/uv -admin -start
```

*dshome* is the DataStage server engine home directory.

## Connecting to UniData Data Sources

If you encounter an error when connecting to a UniData data source, such as:

UniData Client error: call to UniOpenPos returned 45 - Client version (11) and server version (12) are incompatible

you need to edit the UNIAPI.INI file in the Windows directory on your client machine to change the value of the PROTOCOL variable to match the server version. So, in the case of the example error, you would need to change the variable value from 11 to 12:

**PROTOCOL = 12**



# Index

## A

- accessing
  - ODBC databases from UNIX servers 3-9
  - remote files via NFS from UNIX servers 3-11
  - remote UniVerse systems from UNIX servers 3-7
- Add Project dialog box 1-6
- Add Tool dialog box 1-37
- adding
  - data sources to *uvodbc.config* 3-9
  - entries to Tools menu 1-37
  - projects 1-5
  - separators in Tools menu 1-38
- Administrators group 1-1
- Ascential DataStage program folder,
  - Administrator 1-2
- at* command 5-4
- authorization details 1-31
- auto-purge defaults 1-12

## C

- capturing meta data 1-14
- Change License Details dialog box 1-31
- changing
  - license details 1-31
  - locales 4-4
  - maps 4-2
  - Tools menu order 1-38
  - user group assignments 1-16
- changing client license 1-32
- character types 4-4
- Cleanup Resources option 1-10
- Clear Status File option 1-10
- client to server connection 5-6
- client/server maps 4-6
- code page 4-7

- Command Interface dialog box 1-35
- Command Output window 1-36
- commands
  - at* 5-4
  - cron* 5-4
  - DataStage Engine 1-35
  - DSPackagePath 2-6
  - relink.uvlibs* 5-6
- Copy button 5-1
- cron* command 5-4
- cron.allow* file 5-5
- cron.deny* file 5-5
- currency signs 4-4
- Customize dialog box 1-37
- customizing the Tools menu 1-37

## D

- data sources
  - adding to *uvodbc.config* on UNIX servers 3-9
  - connecting to UniData 5-7
  - defining ODBC 2-3, 3-9
- DataStage Administration window 1-4, 4-2
  - General page 1-2
  - Projects page 1-3
- DataStage Director
  - enabling job administration in 1-10
  - Job Schedule view 5-2
- DataStage Engine
  - issuing commands 1-35
- DataStage engine configuration on UNIX servers
  - starting and stopping 3-6
- DataStage Engine Resource Service 2-3
- DataStage licenses 1-31
- DataStage Repository Import dialog box 1-8
- DataStage Telnet Service 2-3

DataStage users, setting up 1–15  
 dates 4–4  
 defining ODBC data sources 2–3, 3–9  
 deleting projects 1–7  
 deployment systems 1–30  
 dialog boxes  
   Add Project 1–6  
   Add Tool 1–37  
   Change License Details 1–31  
   Command Interface 1–35  
   Customize 1–37  
   DataStage Repository Import 1–8  
   Export 1–8  
   User Rights Policy 5–3  
 documentation conventions iv  
*dsadm* user 1–1  
 DSPackagePath command 2–6  
*dsr\_sched.log* file 5–2  
 dsrpc daemon 5–6  
 DSRPC Service 2–3  
 dsrpcd daemon 3–5

## E

editing *uvodbc.config* 2–4, 3–7  
 enabling  
   job administration in Director 1–10  
   tracing on the server 1–17  
 enabling editing of internal reference  
 information 1–12  
 errors in DataStage 5–1  
 Export dialog box 1–8

## F

file units  
   running out of 3–4  
 files  
   cleaning up 1–10  
   *cron.allow* 5–5  
   *cron.deny* 5–5  
   *dsr\_sched.log* 5–2  
   *lib* 5–6  
   log 1–11, 1–12  
   remote 3–11  
   trace 1–17  
   *uvodbc.config* 2–4, 3–7, 3–9

## G

General NLS Settings window 4–6

## I

installation requirements 3–2  
 installing  
   DataStage packages 2–5, 3–10  
   locales 4–5  
   maps 4–7  
   MetaBrokers 2–7  
   plug-ins 2–5  
 issuing DataStage Engine commands 1–35

## J

job administration 1–10  
 job properties, mainframe 1–20  
 Job Schedule view 5–2  
 jobs  
   cleaning up files 1–10  
   clearing status file 1–11  
   log files 1–11  
   scheduling 1–19, 2–2, 3–11, 5–2  
   termination problems 5–5

## K

kernel parameters 3–4

## L

*lib* files 5–6  
 license  
   changing client 1–32  
   upgrading client 1–34  
 license details, changing 1–31  
 locales 4–4  
   changing 4–4  
   installing 4–5  
 locator tables 1–14  
 log files  
   automatic purging 1–11  
   manual purging 1–12  
   setting up operator's view 1–17

## M

mainframe job properties, setting 1–20  
 Mainframe page 1–21  
 maps 4–2  
   changing 4–2  
   client/server 4–6  
   installing 4–7  
 message handlers 1–25  
 meta data 1–14, 2–7  
 MetaBrokers, installing 2–7  
 MetaStage 1–14

moving projects 1-7

## N

National Language Support, *see* NLS

NFS, accessing remote files via (UNIX servers) 3-11

NLS (National Language Support)

client/server maps 4-6

locales 4-4

maps 4-1

overview 4-1

NLS option

project settings 4-2

server settings 4-6

NT system authority 1-19

numbers 4-4

## O

ODBC configuration file 2-4, 3-7

ODBC connection problems 5-5

ODBC data sources, defining 2-3, 3-9

ODBC drivers 5-5

Omit option 3-10

## P

Package Installer 2-5, 3-10

command line options 2-6

Permissions page 1-16

plug-ins, installing 2-5

primary domain server user groups 1-17

process meta broker 1-14

Project NLS Settings window 4-2

Project Properties window 1-9, 1-16, 1-18, 1-19, 1-21

project properties, editing 1-9

projects

adding 1-5

deleting 1-7

moving 1-7

properties

mainframe 1-20

project 1-9

server-wide 1-4

purging, automatic 1-12

## R

radix 4-4

relicensing DataStage 1-31

*relink.uvlibs* command 5-6

remote deployment 1-30

remote files 3-11

remote UniVerse systems 2-4

requirements

installation on UNIX server 3-2

UNIX environment for *dsrpcd* 3-5

restarting

UNIX server engine 3-6

Windows server engine 2-2

restrictions on UNIX servers

Omit option 3-10

scheduling jobs 3-11

*root* user 1-1

## S

schedule log 5-2

Schedule page of Project Properties

window 1-19

scheduled jobs, troubleshooting 5-2

scheduling jobs 1-19, 2-2, 3-11, 5-2

server engine

starting and stopping 2-2

starting and stopping from UNIX 3-6

starting and stopping from Windows 2-2

server timeout, setting 1-4

server tracing 1-17

server-wide properties, setting 1-4

setting

auto-purge defaults 1-12

mainframe job properties 1-20

server timeout 1-4

server-wide properties 1-4

user name for scheduled jobs 1-19

setting up DataStage users 1-15

shared libraries 5-6

sort order 4-4

starting the server engine

from UNIX 3-6

from Windows 2-2

starting the Windows Schedule service 2-2

stopping the server engine

from UNIX 3-6

from Windows 2-2

symbolic links 5-6

## T

testing user name for scheduled jobs 1-20

thousands separator 4-4

times 4-4

Tools menu

adding a separator 1-38

adding an entry 1-37

- changing order of entries 1-38
- customizing 1-37
- trace files, viewing 1-18
- tracing server activity 1-17
- troubleshooting scheduled jobs 5-2
- tuning UNIX parameters 3-4

## U

- Unicode 4-1
- UniData data sources, connecting 5-7
- UniVerse
  - accessing remote systems 3-7
  - remote systems 2-4
- UniVerse systems 2-4
- UNIX kernel parameters 3-4
- UNIX servers
  - and scheduling 5-4
  - restrictions 3-10
- upgrading client license 1-34
- user categories 1-15
  - assigning user groups 1-16
- User Rights Policy dialog box 5-3
- USS systems 1-27
- uvodbc.config* file
  - adding data sources 3-9
  - editing 2-4, 3-7

## V

- View trace file window 1-18
- viewing
  - locales 4-4
  - maps 4-2
  - trace files 1-18
- views, job schedule 5-2

## W

- windows
  - Command Output 1-36
  - DataStage Administration 1-2, 1-4, 4-2
  - General NLS Settings 4-6
  - Project NLS Settings 4-2
  - Project Properties 1-9, 1-16, 1-18, 1-19, 1-21
  - View trace file 1-18
- Windows Schedule service 1-19, 2-2, 5-2
- Windows servers and scheduling 5-2

## Z

- Z/OS systems 1-27
- z/OS systems 1-27