

Ascential **DataStage**

Install and Upgrade 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

This manual guides you through installing or upgrading DataStage[®]. You do not need to read all of it, read those chapters appropriate to the type of install you are doing and the platform on which you are installing.

To find particular topics 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 Guide

This guide contains the following:

- [Chapter 1](#) provides an overview of the installation process and which chapters you need to read.
- [Chapter 2](#) describes installing DataStage for the first time on a Windows platform.
- [Chapter 3](#) describes installing DataStage for the first time on a UNIX platform.
- [Chapter 4](#) describes upgrading an existing installation of DataStage on a Windows platform.
- [Chapter 5](#) describes upgrading an existing installation of DataStage on a UNIX platform.
- [Chapter 6](#) describes configuration that needs to be done if you have installed DataStage Enterprise Edition.

- [Chapter 7](#) describes installing component on a mainframe platform when you have installed Enterprise MVS Edition.
- [Chapter 8](#) describes setting up a USS system for running parallel jobs.
- [Chapter 9](#) describes what's new in the release.
- [Chapter 10](#) gives troubleshooting information.
- [Appendix A](#) gives information needed if you are upgrading from a release earlier than the immediately preceding one.

Documentation Conventions

This manual uses the following conventions:

Convention	Usage
Bold	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.
Lucida Typewriter	In examples, Lucida Typewriter bold indicates characters that the user types or keys the user presses (for example, <Return>).
[]	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 File ➤ Exit " means you should choose File from the menu bar, then choose Exit from the File pull-down menu.

Convention	Usage
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 *Install and Upgrade Guide***. This guide contains instructions for installing DataStage on Windows and UNIX platforms, and for upgrading existing installations of DataStage.
- **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 NLS Guide.** This Guide contains information about using the NLS features that are available in DataStage when NLS is installed.

These guides are 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 Guide	ix
Documentation Conventions	x
DataStage Documentation	xi

Chapter 1 Introduction

Using DataStage Manuals and Help.....	1-2
Using the Adobe Acrobat Reader.....	1-2

Chapter 2 Installing on Windows Systems

Pre-Install Checks.....	2-1
Server	2-2
Enterprise MVS Edition	2-2
Enterprise Edition.....	2-2
DataStage Packs.....	2-3
Plug-Ins	2-3
Hardware and Software Requirements	2-3
Server	2-4
Client	2-4
Disk Space.....	2-4
Networking	2-5
Media.....	2-5

The Installation Process	2-5
Server Install.....	2-6
Client Install	2-12
Systems with UniVerse Installed	2-14
Changing the User who Runs the DataStage Services.....	2-15

Chapter 3

Installing on UNIX Systems

Types of Install.....	3-1
Root Install	3-1
Non-Root Install	3-2
UNIX Install Checklist	3-3
Server	3-4
Enterprise MVS Edition	3-4
Enterprise Edition.....	3-4
DataStage Packs.....	3-4
Plug-Ins	3-5
Contents of the CD-ROMs	3-5
Pre-Install Checks and Set Up.....	3-6
Setting Up Users and Group.....	3-6
Decide on Installation Directory	3-7
Select Suitable Shell	3-7
Set Kernel Parameters.....	3-7
Pre-Install Special Considerations	3-10
Hardware and Software Requirements	3-12
Server	3-12
Disk Space	3-12
Enterprise Edition.....	3-13
USS Systems	3-13
C++ Compilers and Run-Time Libraries.....	3-13
Client.....	3-15
Networking	3-15
Media	3-15
Installing DataStage Server	3-15
Reinstalling and Maintenance Menu	3-22
Installation Log Files	3-23
Installing DataStage Clients	3-23

Post-Install Checks and Configuration	3-26
Non-Root Installs	3-26
Tuning the DataStage Server Engine	3-26
Environment Variables.	3-28
Environment Variables for Enterprise Edition	3-29
Configuring Enterprise Edition	3-29
Configuring for Use with PAM	3-29
Configuring Plug-ins	3-30
Configuring ODBC Access	3-32
Configuring for NLS Use	3-39
Upgrading Non-Root Installations	3-39

Chapter 4
Upgrading Windows Systems

Upgrade Issues.	4-1
Job Executables	4-1
XML Pack.	4-1
The Upgrade Process.	4-2
Server Upgrade	4-2
Client Upgrade	4-6

Chapter 5
Upgrading UNIX Systems

Types of Upgrade.	5-1
Root Upgrade	5-2
Non-Root Upgrade	5-2
Upgrade Issues.	5-3
Job Executables	5-3
XML Pack.	5-3
UNIX Upgrade Checklist	5-4
Pre-Upgrade Checks and Set Up	5-5
Platforms Supported by DataStage Release 7.5.1	5-5
Select Suitable Shell	5-5
Adding the dsadm User.	5-5
ODBC Configurations	5-6
Back Up Files.	5-6
System Requirements	5-7

Upgrading DataStage	5-7
Reinstalling and Maintenance Menu	5-13
Installation Log Files	5-15
Upgrading DataStage Clients	5-15
Post-Upgrade Checks and Configuration	5-18
Non-Root Upgrades	5-18
DataStage Users, UNIX Permissions and Security	5-19
Configuring for Use with PAM	5-19
Environment Variables	5-20
Configuring Plug-ins	5-22
Configuring ODBC Access	5-23
Upgrading Non-Root Installations	5-30

Chapter 6

Configuring for Enterprise Edition

Your System's Remote Shell	6-1
Specifying the Location of rsh	6-1
Allowing User Execution of rsh Without a Password	6-2
Copying the Parallel Engine to Your System Nodes	6-3
Setting the wall System Parameter	6-4
Setting the Maximum Number of Processes	6-4
Setting the Maximum Amount of Memory on AIX	6-5
Setting the Per-User Resource Limits on Compaq Tru64	6-6
Increasing the Limits on Segment Sizes on HP-UX	6-7
Setting the System Paging Space	6-8
Setting the Paging Space on AIX	6-8
Setting the Paging Space on Sun Solaris	6-9
Setting the Paging Space on Compaq Tru64	6-9
Setting the Paging Space on HP-UX	6-10
Setting the Paging Space on LINUX	6-10
Edit /etc/magic	6-11
Setting Environment Variables	6-11
Temporary Directory	6-11
Network Settings	6-11
Transform Library	6-12
Job Monitoring	6-12
Setting Transport Block Size	6-12
Dumping Information	6-12

C++ Compiler	6-12
Configuring the Parallel Environment	6-13
Granting User Privileges to run Parallel Jobs	6-13
Upgrading from Beta Releases	6-14
Setting Up Database Connectivity	6-14
DB2/UDB Databases	6-14
Oracle Databases	6-17
Teradata Databases	6-19
Informix XPS Databases	6-20

Chapter 7
Installing DataStage Components on a Mainframe Platform

Installing the Copylib Members	7-2
Installing the Runtime Library (RTL)	7-2
Running the Mainframe IVP	7-4

Chapter 8
Installing DataStage Components on a USS System

Requirements	8-1
System Requirements	8-1
Hardware Requirements	8-2
Development Environment	8-2
Connectivity	8-2
Running the Install	8-2
Configuring the System	8-4
Remote Shell Capability	8-4
Specifying the Location of rsh	8-5
Allowing User Execution of rsh Without a Password	8-5
Setting Environment Variables	8-6
Location of Parallel Engine	8-6
Location of Configuration File	8-7
Temporary Directory	8-7
Job Monitoring	8-7
Setting Transport Block Size	8-7
C++ Compiler	8-7
Setting Up DB2 Access	8-8

Chapter 9 New Features in DataStage

What's New at Release 7.5.1	9-1
SQL Builder	9-1
Interaction with MetaStage	9-1
USS Support	9-1
Remote Deployment	9-2
Enhancements to Job Reporting	9-2
Multi-job Compiler	9-2
Enhancements to Command Line and API Facilities	9-2
Specifying Icons for Custom Stages	9-2
Vector Support in Parallel Transformer Stages	9-2
Editing of Multiple Derivations Columns	9-2
What's New at Release 7.5	9-2
Job Sequencer	9-3
Mobile Director	9-3
USS Support	9-3
Improved Designer Interface	9-3
Client Licenses	9-3
Job Reporting	9-3
Multi-Client Manager	9-3
Visual Cues	9-3
Parallel Job Message Handlers	9-4
Parallel Job Lookup Stage	9-4
Previous Enhancements	9-4
New DataStage Server	9-4

Chapter 10 Troubleshooting

Troubleshooting UNIX Installations	10-1
Running Out of File Units	10-1
Running Out of Memory on AIX	10-2
dsrpcd Issues	10-2
ODBC Connection Problems	10-3
Problems Using Oracle 8 CLI Plug-ins	10-3
Login Problems on Windows Systems	10-4

Problems and Restrictions	10-4
Stored Procedures Using ODBC - UNIX	10-4
Connecting to the Server - UNIX	10-5
Object Call Install - UNIX	10-5
Deadlock Daemon - Windows and UNIX	10-5
Using ODBC stage to access Sybase on Compaq Tru64	10-5
Using Oracle 8 OCI and ODBC Stages in the Same Job	10-5

Appendix A

Upgrading Issues

General Install Issues	A-1
Coresident UniVerse Systems – Windows Systems	A-1
Coresident Systems – UNIX Systems	A-2
Verifying Coresident Installs	A-3
XMLPack and ClickPack	A-4
Oracle OCI 9 Plug-in Renamed	A-5
Creating MetaStage Locator Tables	A-5
Capturing Process Meta Data - Locator Table setup	A-7
Windows Install Issues (Client and Server)	A-7
RTI Server must be stopped manually before running install	A-7
Windows 2000: Access required to the temp directory	A-8
Windows 2000: Login Policy must be set	A-8
Failure to Register .ocx on Some Systems	A-8
Windows 2000 Advanced Server: Importing Active/X Routines	A-9
External Active-X Routine arguments	A-9
Windows 2000 Adv Server: Reporting Assistant update failure	A-10
Documentation Tool Install Issues	A-10
Use of Virus Scanners on the DataStage Server	A-11
UNIX Install Issues	A-11
Administrative User	A-11
The dsenv File	A-11
Reverting to a Previous Version of the DataDirect ODBC Drivers	A-12
Change to Operation of dsjob on UNIX	A-12
Solaris Install Issues	A-13
AIX Install Issues	A-13
HP-UX Install Issues	A-14
Compaq Tru 64 Install Issues	A-14

Problems and Restrictions - Release 7.0, 7.0.1. and 7.1 Onwards.	A-14
Time Based Job Monitoring - Intermittent Problems	A-15
Performance Improvement for Enterprise Edition on Tru 64 Systems . .	A-15
Oracle 7 OCI Plug-in Not Supported.	A-15
Oracle OCI8i Plug-in No Longer Supports Oracle 8.1.6 on AIX	A-15
Using Oracle 9i or 10g in Parallel Jobs on HP-UX	A-15
LD_LIBRARY_PATH library path order for Oracle 8i BulkLoader Plug-in on TRU64	A-16
AxcelPack	A-16
Problems and Restrictions – Release 6.0 Onwards.	A-16
UniData Stage - Fails to write to a UniData Table.	A-16
DataStage Client/Server RPC Inactivity timeout	A-16
Setting SHMMNI on HP-UX Systems (UNIX)	A-17
Problems and Restrictions – Release 5.2.2 Onwards	A-17
Defining Connections to Remote UniVerse databases.	A-17
Change to Input Stage Delimited Sequential Files	A-17
Problems and Restrictions – Release 4.2.2 Onwards	A-18
Hashed File Stage Modifications	A-18
Recommended not to use LAN Manager connectivity for import.	A-18
Enhanced Transaction Grouping Support.	A-18
DataStage Signal Handling Issue on Solaris (UNIX).	A-19
Oracle OCI 8 Plug-in Support for Operating System Authentication. . . .	A-20
Oracle OCI 8 Plug-in Pre 4.2 User-defined SQL behavior	A-20
Problems and Restrictions – Release 4.2 Onwards.	A-21
Installation of Developer Studio onto DataStage clients (Windows). . . .	A-21
Defining Sequential File Meta Data	A-22
Red Brick ODBC Driver Message Suppression (Windows).	A-22
Packager Installer log files.	A-22
Modified Sequential File stage handling of Numeric & Decimal.	A-22
DataStage Enterprise MVS Specific Issues	A-23
Release 7.x Enhancements	A-23
Release 6.0 Enhancements	A-23
Release 6.0 Problems and Restrictions	A-26
Release 5.2.2 Enhancements.	A-26
Release 5.2.2 Problems and Restrictions	A-27
Release 5.2 Problems and Restrictions	A-28
Release 5.1 Problems and Restrictions	A-28

DataStage Enterprise Edition Specific Issues	A-29
Recompiling Parallel Jobs	A-29
Regeneration of Filesets	A-29
Change to Final delimiter behavior for Parallel jobs.....	A-29
SyncSort Option Removed from Parallel Sort Stage	A-30
DataStage Server Job Specific Issues	A-30
Transform Routine Arguments	A-30
New CRC32() Function Available	A-31
Active to Active link Row Buffering - Optional Performance Enhancement Introduced at Release 5.1	A-32
UniVerse and Hashed File Stages - Compatibility between Releases ...	A-32
Hash File Write Errors	A-33
UniData 6 Stage	A-33

1

Introduction

This guide tell you how to install DataStage for the first time and how to upgrade from an earlier version of DataStage to release 7.5.1.

Read the chapter that applies to your situation:

- If you are installing DataStage for the first time on a Windows system, read [Chapter 2](#).
- If you are installing DataStage for the first time on a UNIX system, read [Chapter 3](#).
- If you are upgrading an existing DataStage installation on a Windows system, read [Chapter 4](#). (If you are upgrading from an earlier release than 7.0, you should also look at [Appendix A](#).)
- If you are upgrading an existing DataStage installation on a UNIX system, read [Chapter 5](#). (If you are upgrading from an earlier release than 7.0, you should also look at [Appendix A](#).)
- If you have installed the DataStage Enterprise Edition, read [Chapter 6](#).
- If you have installed the DataStage Enterprise MVS Edition, read [Chapter 7](#)
- If you are intending to run parallel jobs on USS systems, read [Chapter 8](#)

[Chapter 9](#) describes those features that are new to DataStage at Release 7.5.1

[Chapter 10](#) gives troubleshooting advice about DataStage 7.5.1.

[Appendix A](#) contains information that is useful if you are upgrading from a previous release of DataStage earlier than 7.5.

Using DataStage Manuals and Help

To read the DataStage manuals from any DataStage client, click the **DataStage Documentation** icon in the Ascential DataStage program folder.

You can access online Help from any DataStage client by choosing **Help** from the menu, or by clicking a **Help** button.

Using the Adobe Acrobat Reader

The Adobe Acrobat Reader has its own online documentation. We do not provide technical support for the Adobe Acrobat Reader. For information, see the Adobe Web site at <http://www.adobe.com>.

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).

2

Installing on Windows Systems

This chapter tells you how to install DataStage on a Windows system for the first time. (If you are upgrading an existing DataStage installation, read [Chapter 4, "Upgrading Windows Systems,"](#) if you are installing on a UNIX system, read [Chapter 3, "Installing on UNIX Systems."](#))

You can install DataStage server and clients on the same Windows systems, or separate ones. You may want to install the client software on a number of systems, connecting to a server on a standalone system. All these permutations are possible.

Pre-Install Checks

Before you start the installation, take a few minutes to prepare:

- Check that your client and server computers match the requirements (see ["Hardware and Software Requirements"](#) on [page 2-3.](#))
- Check that you have authorization details for the DataStage server and client software. You must specify the correct authorization details for the component you are installing, according to your licensing agreement.
- Check that you have sufficient access rights. You need full access rights to the directory where you want to install the client software and be logged on as a Windows Local Administrator. To install the Windows server software you must log on as a Windows Local Administrator on the server computer. You can, if you want, create a user to install and administer DataStage. This user should have full administrative rights. (Once you have installed DataStage you

should use the DataStage Administrator to set up the user roles – see "[User Roles on Windows Systems](#)" in *DataStage Administrator Guide*.)

- Decide where you want to install DataStage on both the client and server computers. A default location is given, but you can browse for another location during the installation, if required.
- Ensure that you have a valid network connection when attached to a domain.

During the server installation you are asked what language you want to install. When you choose English (the default), you are asked if you want to install National Language Support (NLS). NLS is installed automatically with any other languages. You are also asked to input authorization codes for the various DataStage Editions – you can provide some or all of these codes.

Server

Server is the version of DataStage that allows you to run jobs on the server using the DataStage engine. If you have this edition, the authorization code you enter during install informs the DataStage install program that the server job type is required.

Enterprise MVS Edition

Enterprise MVS Edition is the version of DataStage that allows you to develop jobs to run on mainframe computers. If you have purchased Enterprise MVS Edition the authorization code you enter during install informs the DataStage install program that the mainframe job type is required.

Enterprise Edition

Enterprise Edition is the version of DataStage that allows you to develop parallel jobs. These run on DataStage UNIX servers that are SMP, MPP, or cluster systems, but you can install it on an Windows server in order to develop jobs which can subsequently be run on a UNIX server.

If you have purchased Enterprise Edition, the authorization code you enter during install informs the DataStage install program that the parallel job type is required.

DataStage Packs

There are a number of packs available with DataStage that extend functionality of DataStage jobs, each providing a set of Plug-in stages and associated functionality.

- **XML Pack.** This package is supplied with DataStage. It provides tools that enable you to convert data between XML documents and data tables. Features and functionality are fully described in the *XML Pack Designer Guide*.
- **Java Pack.** This is an add-on package. It comprises two template stages and an API which enables you to implement DataStage stage in Java. It is described in the online documentation accompanying the pack.
- **RTI Agent.** This is separately licensed. It allows you to publish DataStage jobs as Web Services. These facilities are described in the *Web Services Designer Guide*.
- **Click Pack.** This is an add-on package. It provides tools that enable you to extract and transform data from Web server log files and email servers. Click Pack facilities are described in the *Click Pack Guide*.

The following separately licensed pack extends the functionality of parallel jobs:

- **SAS Integration pack.**

The following separately licensed pack extends the functionality of mainframe jobs:

- **MVS IMS Source.**

Plug-Ins

Plug-ins are specialized packages that allow you to connect to specific databases or perform other special functions. Once installed, the Plug-ins provide new stage types for use in your jobs. Versions are provided for use with server jobs or parallel jobs.

Hardware and Software Requirements

Check that you have the following hardware and software before you install DataStage.

Server

A PC with an Intel processor (or equivalent) running Windows 2000 service pack 4 or later (Professional, Server, Advanced Server), or Windows 2003 Server (Standard, Enterprise). A minimum of 256 MB memory.

Client

A PC with an Intel processor (or equivalent) running , Windows 2000 service pack 4 or later (Professional, Server, Advanced Server), or Windows XP Professional Service Pack 1. Microsoft Internet Explorer 5.01 SP2 or better. A screen resolution of 800 x 600 or better is recommended with True Color (24-bit). A minimum of 128 MB memory.

If you want to use the same computer as both client and server, you must use a PC with an Intel processor (or equivalent).

You also require .NET framework v1.1.

Disk Space

You need the following disk space:

On the server:

- 80 MB to install the DataStage server.
- 4.5 MB per project to install the DataStage server.
- 2.2 MB in the Windows system32 directory.
- 90 MB for RTI Plug-ins and JVM.
- Sufficient storage space for any data to be held in DataStage tables or files.

On the client:

- 10 MB in the Windows 2000 or Windows XP system32 directory, as appropriate.
- 200 MB in the DataStage install directory for DataStage clients and documentation (typical installation).
- Additional space to allow for temporary data storage while a DataStage job is running.

Networking

The client and server should be networked using TCP/IP network protocol or LAN Manager.

When installing, you should ensure you have a valid network connection when attached to a domain. If your primary domain controller (PDC) cannot be found across the network, set up will fail when licensing the DataStage engine. If you experience problems you should install the product using a local administrator on the machine (workgroup users are not affected).

Media

Both your client and your server need access to a CD-ROM drive.

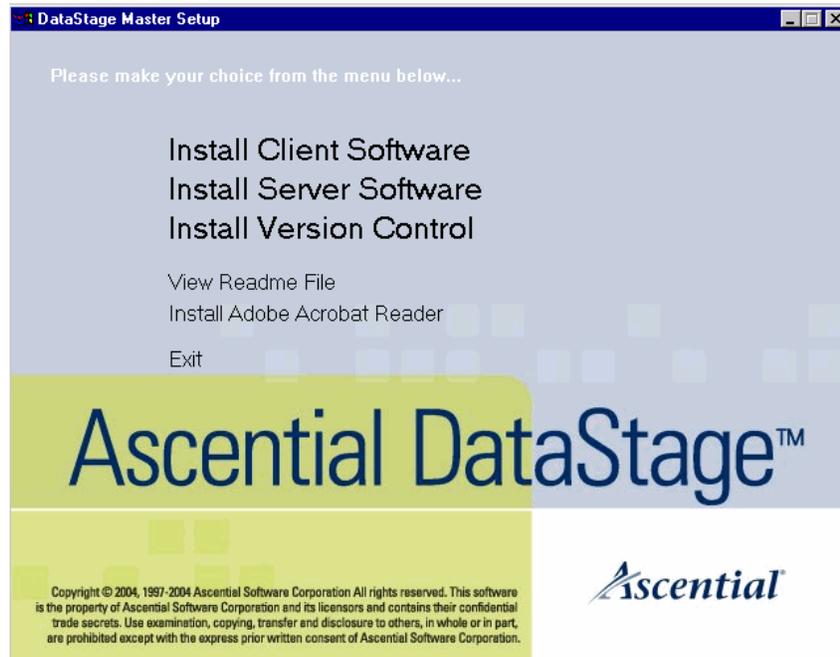
The Installation Process

The installation for DataStage client components is common to all platforms, but the DataStage server installation differs according to the type of server:

- For Windows servers, use these instructions.
- For UNIX servers, use the separate instructions supplied in [Chapter 3](#).

The DataStage CD-ROM contains a master setup program that manages the installation. To view the DataStage Master Setup screen, insert the CD-ROM in the drive. The screen should appear automatically. If Auto Run is switched off and the screen does not appear, choose the CD-ROM drive from Windows **Explorer**, then

double-click **install.exe**. The DataStage Master Setup screen is shown below.

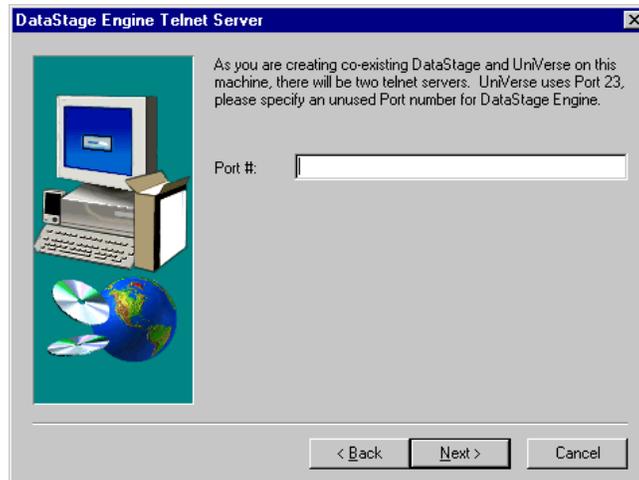


From this screen you can install the DataStage client software, server software for Windows servers, and the Adobe Acrobat Reader, which is used to view the DataStage online documentation. You can also install DataStage Version Control (see *Version Control Guide*). Whichever option you choose, a setup program guides you through the process.

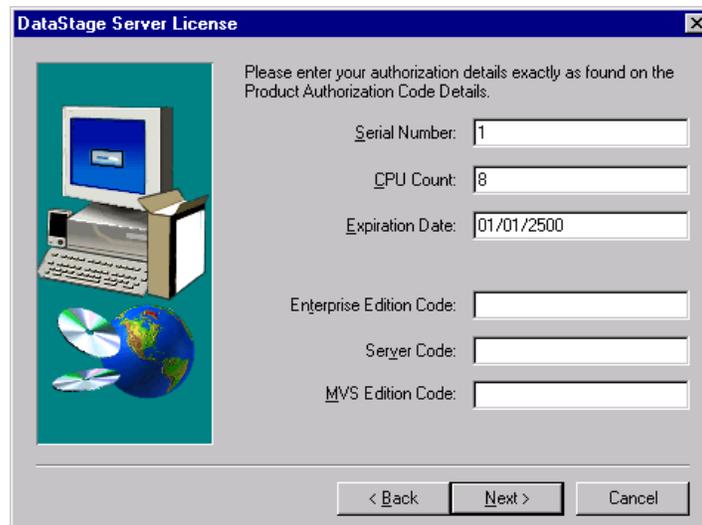
Server Install

The server install process is follows:

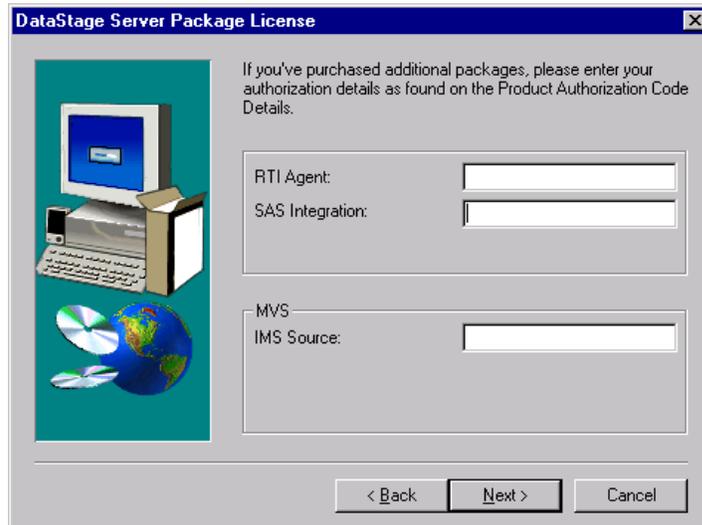
- 1 If you have an installation of UniVerse on your Windows platform, you are asked to specify a new port number (see "Systems with UniVerse Installed" on page 2-14).



- 2 You are asked to enter your authentication details. The installation checks that the number of CPUs licensed is sufficient for your system. Enter Serial Number, CPU Count, and Expiration Date as they appear on your license. Enterprise Edition, Server, and Enterprise MVS Edition have separate codes. Enter the codes for the version or versions you have purchased.



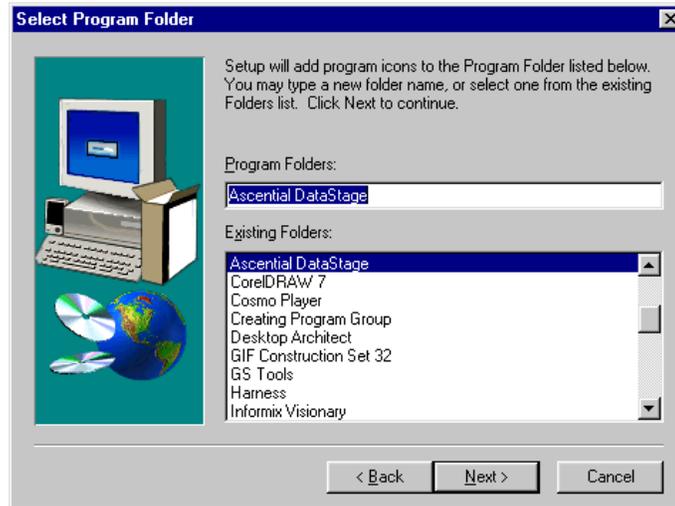
- 3 You are asked to input authorization codes for the optional packages. Click **Next** when you have entered the codes for the packages you have purchased.



- 4 You are asked to accept the default installation directory (c:\Ascential\DataStage) or specify a new one.



- 5 You are asked to accept the default Program folder (Ascential DataStage) or specify a new one.

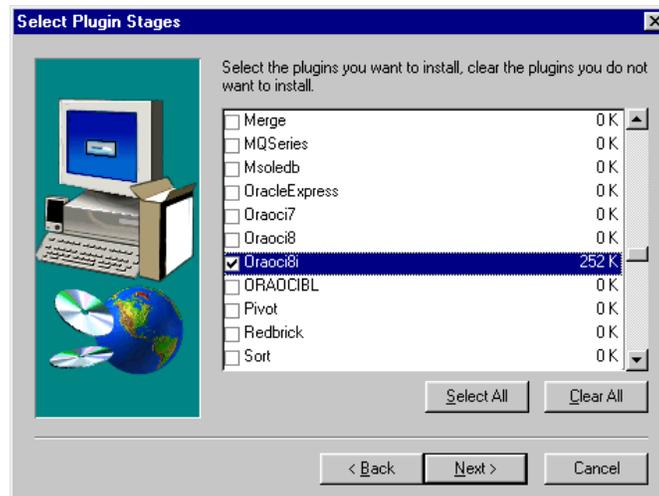


- 6 You are asked to choose a language. If you choose English (the default) the next screen will ask you whether you want to install NLS. NLS is automatically installed if you select any other language.

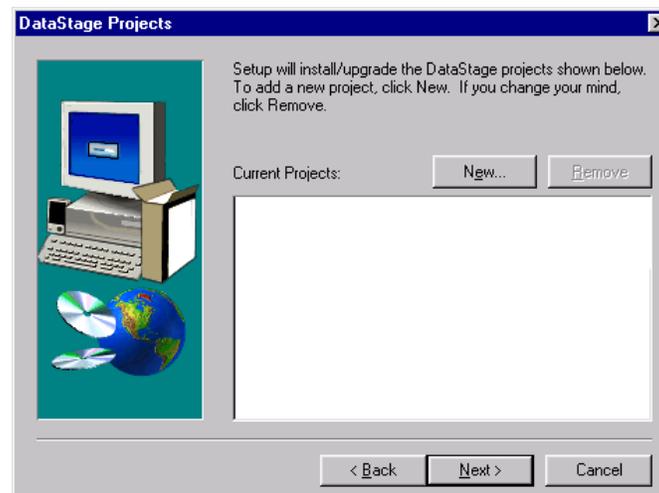


- 7 You are asked to select the Plug-ins you want to install. Plug-ins are specialized packages that allow you to connect to specific databases or perform other special functions. You can install them at any time, so don't worry if you miss one you need at this stage (see [Server Job Developer's Guide](#) for more information on Plug-

ins – information about individual Plug-ins can be found in the DataStage Packages\PlugIn Stages\Docs\ENU directory on the install CD-ROM).



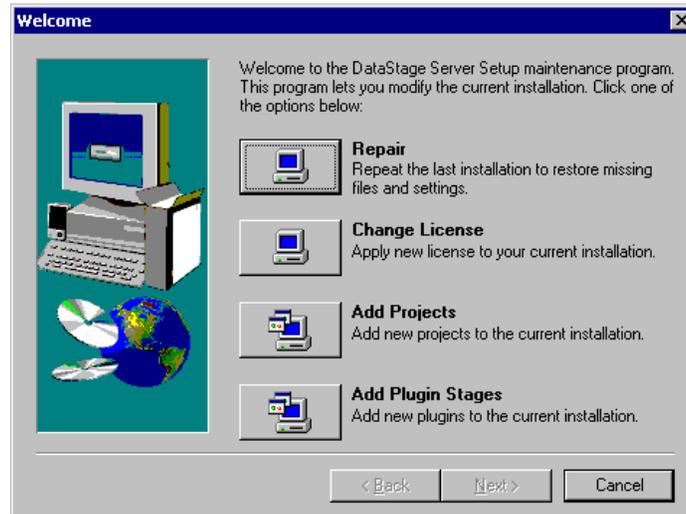
- 8 You are asked to specify the names of projects you want to set up. You must specify at least one project at this stage. Click **New...** and type in a project name for each project you want to add. (You can add protected projects at this point if required, see "[Protecting a Project](#)" in *DataStage Administrator Guide* for details about protected projects).



The server install then proceeds. When it finishes it asks if you want to view the release notes and/or start the DataStage Services (you need to start the services before you can run DataStage).

You can repeat the install to run the setup program in maintenance mode. This allows you to repeat the previous install, or add new projects or Plug-ins, or relicense to add new capacity or functionality.

Choose an option and the setup program guides you through the process.

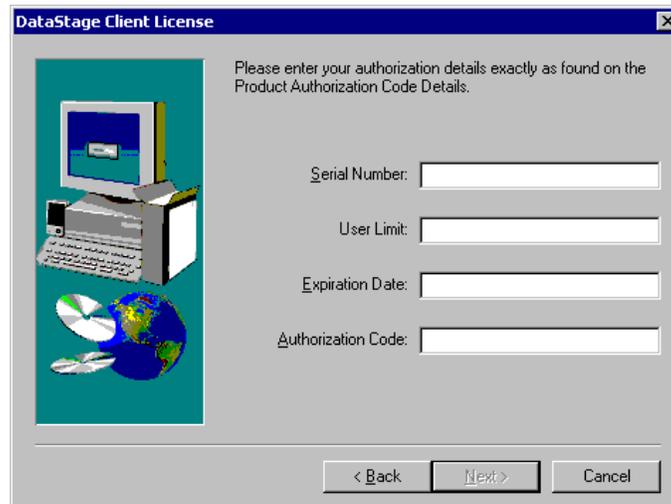


After installing the client and server software, check that the DataStage configuration meets your requirements. For information about configuring DataStage, see *DataStage Administrator Guide*.

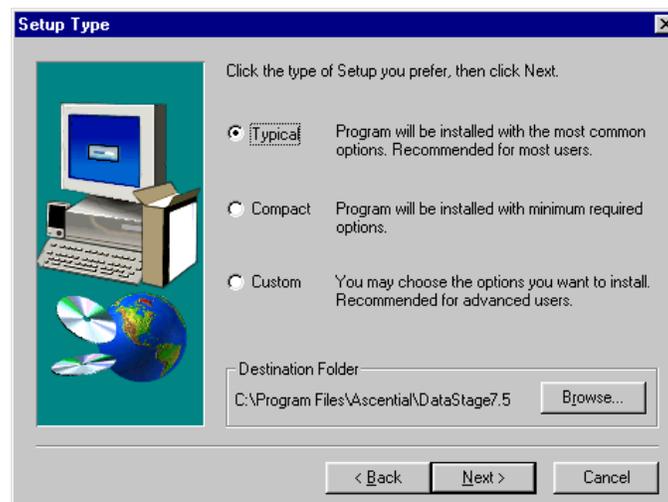
Client Install

The client install process is as follows:

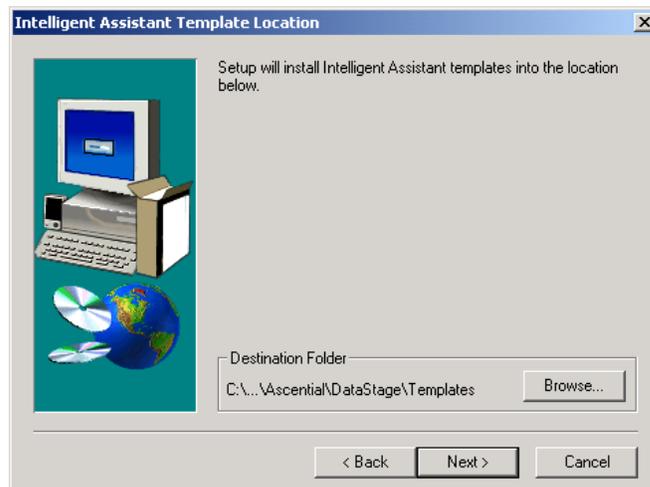
- 1 You are asked to enter your authentication details.



- 2 You are asked to accept the default installation directory (c:\Program Files\Ascential\DataStage7.5.1) or specify a new one and to choose between **Typical**, **Compact**, and **Custom** installs (**Typical** is recommended).



- 3 A screen asks you where you would like to install the Intelligent Assistant templates. Accept the default destination or browse for a new one.



The client install then proceeds. When it is finished it asks if you want to add shortcuts to your desktop, and whether you want to view the release notes.

If you intend to use the DataStage reporting facilities, you need the Documentation Tool. It is installed automatically if you select **Typical Installation** from the client setup program (see step 2). You should already have installed the Microsoft ODBC driver manager and the Microsoft Access ODBC driver on the DataStage client. The online Read Me file contains the latest news about this release of DataStage, including Documentation Tool requirements.

Systems with UniVerse Installed

If you are installing onto a system which has a UniVerse database installed, you need to take special steps to install DataStage as a coresident system during the installation of the server component.

The setup program asks you to specify a new port number for the DataStage server (UniVerse retains port 23) – see step 1 on [page 2-7](#). You can select any currently unused port. The `Winnt\System32\drivers\etc\Services` file gives a list of all currently used ports. Installing the DataStage server adds an entry called `dstelnet` which you can subsequently edit if required. If you are unsure about port numbers, ask your system administrator.

Once a port number is selected and you select Next, the setup continues.

Changing the User who Runs the DataStage Services

By default a Windows server runs the DataStage services using the LocalSystem account, but you can set up another user to run the services if required.

To do this:

- 1 Open the **Services** dialog box in the control panel.
- 2 In the **Properties** dialog box for each of the DataStage services, go to the **Log On** tab.
- 3 Under **Log on as:** select the **This account** option, specify the user and supply the password.

Note that the user you choose must have the following privileges:

- logon local
- act as part of the OS
- replace a process level token
- create a token object.

The DataStage services are:

- DSRPC
- DataStage Engine Resource
- DataStage Telnet

3

Installing on UNIX Systems

This chapter tells you how to install DataStage on a UNIX system for the first time. (If you are upgrading an existing DataStage installation, read [Chapter 5, "Upgrading UNIX Systems."](#))

The DataStage Server is installed on a UNIX platform. The DataStage clients are installed on a Windows 2000 or Windows XP systems. You can install clients onto several Windows platforms if required.

You should check the Read Me file and "[Problems and Restrictions](#)" on [page 10-4](#) before installing to check for any platform-specific issues that might affect your installation. The read me file can be found in the top level CD-ROM directory and is called *readmeux.txt*.

If you are installing the Enterprise Edition, be sure to read [Chapter 6, "Configuring for Enterprise Edition."](#)

You should be familiar with UNIX system administration in order to perform an install.

Types of Install

There are two basic types of install on a UNIX system: root and non-root.

Root Install

DataStage installed by *root* is installed in impersonation mode with auto start enabled. During installation an administrative user is specified who is authorized to carry out day-to-day administrative tasks. By default this is *dsadm*, but you can specify another user if required.

In impersonation mode, the DataStage server inherits the identification and permissions of the user who has logged onto it. The DataStage server will then execute all functionality using that user's identification and permissions.

Auto start is enabled with impersonation mode, so that the DataStage services start automatically whenever the machine is restarted.

Non-Root Install

DataStage installed by a user other than *root* is installed in non-impersonation mode with auto start disabled. Whoever performs the installation is designated the product administrator.

Under non-impersonation mode, the DataStage server inherits the identification and permissions of the user who started the DataStage services (usually the product administrator). The DataStage server then executes all functionality using this identification and associated permissions, regardless of the actual id of the user logged in.

Because the server runs with the permissions of the administrative user, every user who logs in will be able to perform administrative tasks.

With this version the administrator will need to manually start the DataStage services whenever the machine is restarted; auto start is not available.

It is intended that after a non-*root* install certain steps are taken by *root* to render DataStage fully functional, in particular:

- DataStage must be enabled by the *root* user running the script `$DSHOME/scripts/DSEservices.sh -add`. This adds a service entry to `/etc/services` for DataStage.
- *Root* must take steps to enable impersonation mode – otherwise no one can log in via a DataStage client. This can be done by running the script `$DSHOME/scripts/DSEenable_impersonation.sh`.

A number of scripts are provided which can be run by *root* in order to uprate a non-root installed system to give more functionality. Your system administrator can examine these scripts and assure themselves of their security before uprating the system, see "[Uprating Non-Root Installations](#)" on [page 3-39](#).

A typical scenario for using non-root install would be as follows:

- 1 Perform the non-*root* installation.
- 2 Have *root* enable DataStage by running `DSEservices.sh -add` (this can be run remotely).

- 3 Test basic DataStage functionality on your system by using *dsjob* to run sample jobs from the command line. (*Dsjob* is described in "[DataStage Development Kit \(Job Control Interfaces\)](#)" in *Server Job Developer's Guide* and "[DataStage Development Kit \(Job Control Interfaces\)](#)" in *Parallel Job Advanced Developer's Guide*.)
- 4 Have root update the system for impersonation mode and auto-startup by running the scripts described in "[Upgrading Non-Root Installations](#)" on [page 3-39](#). Again, these can be run remotely.

UNIX Install Checklist

This section provides a checklist of the steps you should take when installing DataStage:

- 1 Check your system meets DataStage installation requirements (see "[Pre-Install Checks and Set Up](#)" on [page 3-6](#)).
- 2 Decide whether you are installing as *root* or as a non-*root* user (see "[Types of Install](#)" on [page 3-1](#)).
- 3 Set up required users and group. In particular you must set up an administrative user (by default called *dsadm*) before installation and a group for the admin user and DataStage users. (See "[Setting Up Users and Group](#)" on [page 3-6](#)).
- 4 Ensure that the user who is going to run the install has an appropriate umask setting. We recommend 022 or 0022.
- 5 Mount the installation CD and run the DataStage install script. Verify install is successful (see "[Installing DataStage Server](#)" on [page 3-15](#)).
- 6 Install DataStage clients on Windows platforms (see "[Installing DataStage Clients](#)" on [page 3-23](#)).
- 7 Perform post-install configuration:
 - a Perform configuration required for any Plug-in stages you have installed with DataStage (see "[Environment Variables](#)" on [page 3-28](#) and "[Configuring Plug-ins](#)" on [page 3-30](#)).
 - b Set up ODBC connectivity for the databases you are reading and writing to using ODBC (see "[Environment Variables](#)" on [page 3-28](#) and "[Configuring ODBC Access](#)" on [page 3-32](#))
- 8 If you wish to use authentication other than standard UNIX authentication, configure DataStage to use PAM (see "[Configuring for Use with PAM](#)" on [page 3-29](#)).
- 9 If you are installing Enterprise Edition, configure your system in accordance with the instructions given in [Chapter 6](#).

- 10 If you are installing Enterprise MVS Edition, install the required components on your mainframe platform in accordance with the instructions given in [Chapter 7](#).
- 11 If you are intending to run parallel jobs on USS systems, install the required components on the USS system in accordance with the instructions given in [Chapter 8](#).

Server

Server is the version of DataStage that allows you to run jobs on the server using the DataStage engine. If you have this edition, the authorization code you enter during install informs the DataStage install program that the server job type is required.

Enterprise MVS Edition

Enterprise MVS Edition is the version of DataStage that allows you to develop jobs to run on mainframe computers. If you have purchased Enterprise MVS Edition the authorization code you enter during install informs the DataStage install program that the mainframe job type is required.

Enterprise Edition

Enterprise Edition is the version of DataStage that allows you to develop parallel jobs. These run on DataStage UNIX servers that are SMP, MPP, or cluster systems.

If you have purchased Enterprise Edition, the authorization code you enter during install informs the DataStage install program that the parallel job type is required.

DataStage Packs

There are a number of packs available with DataStage that extend functionality of DataStage jobs, each providing a set of Plug-in stages and associated functionality.

- **XML Pack.** This package is supplied with DataStage. It provides tools that enable you to convert data between XML documents and data tables. Features and functionality are fully described in the *XML Pack Designer Guide*.
- **Java Pack.** This is an add-on package. It comprises two template stages and an API which enables you to implement DataStage stage in Java. It is described in the online documentation accompanying the pack.

- **RTI Agent.** This is separately licensed. It allows you to publish DataStage jobs as Web Services. These facilities are described in the *Web Services Designer Guide*.
- **Click Pack.** This is an add-on package. It provides tools that enable you to extract and transform data from Web server log files and email servers. Click Pack facilities are described in the *Click Pack Guide*.

The following separately licensed pack extends the functionality of parallel jobs:

- **SAS Integration pack.**

The following separately licensed pack extends the functionality of mainframe jobs:

- **MVS IMS Source.**

Plug-Ins

Plug-ins are specialized packages that allow you to connect to specific databases or perform other special functions. Once installed, the Plug-ins provide new stage types for use in your jobs. Plug-ins are available for server and parallel jobs.

Contents of the CD-ROMs

The server CD-ROM contains DataStage server software for UNIX, which is installed by executing the *install.sh* program, see "[Installing DataStage Server](#)" on [page 3-15](#).

Other software is also present on the CD, each in its own directory. Note that the directory names and file names may not appear exactly as shown below - those longer than eight characters will be truncated to six characters with a number added, for example, *clickp~1*.

- *clickpack*. This directory contains the DataStage ClickPack installable. The file *clickpack_techbull.pdf* describes how to install and use ClickPack.
- *packages*. This directory contains DataStage Plug-ins. You can either install Plug-ins through the Server install or by using the DataStage Package Installer (*dspackinst*) as described in "[Installing DataStage Packages](#)" in *DataStage Administrator Guide*. We recommend that you use server install (if you want to add a new Plug-in after install, re-run install in maintenance mode). This way future DataStage upgrades will automatically upgrade your Plug-ins.

Pre-Install Checks and Set Up

This section details set up steps you have to perform and configuration adjustments you may have to make **before** installing the DataStage server on your UNIX platform.

First you should check your system meets DataStage installation requirements and (see "[Hardware and Software Requirements](#)" on [page 3-12](#)). You should also ensure that you have authorization codes for both server and client.

Setting Up Users and Group

DataStage can be administered on a UNIX platform by a special non-root user. This is *dsadm* by default, but you can specify a different administrative user at install. You must set up this user before installing DataStage. All DataStage users should belong to the same UNIX group, and this should be the administrative user's primary group. We recommend you name the group *dstage*.

If you want to set up the system so that it distinguishes between product managers, developers and operators, you should set up secondary groups for each class of user. Each user is then allocated to the product manager, developer, or operator secondary group (but not to more than one secondary group). You can then use the DataStage Administrator to assign the appropriate DataStage user role to the secondary groups. See "[User Roles on UNIX Systems](#)" in *DataStage Administrator Guide* for details. Operators cannot use the DataStage Designer and only see released jobs in the DataStage Director. Neither Operators nor Developers can create protected projects or add anything to them.

Note On Compaq Tru64 systems, the administrative user must be a member of the system group (group number 0). This is necessary to ensure that they can administer DataStage correctly.

You can, if required, use an authentication mechanism other than the standard UNIX one. To this end, DataStage supports PAM (pluggable authentication modules). This provides a way of keeping underlying authentication technologies separate from application code, thus eliminating the need to update the application every time the authentication mechanism is changed. To use PAM authentication, you need to configure DataStage after you have installed it, see "[Configuring for Use with PAM](#)" on [page 3-29](#).

Decide on Installation Directory

You should decide before installation where you want the DataStage server to reside. The default installation location is `<dsadmhome>/Ascential`, where `<dsadmhome>` is the home directory of the administrative user. You can accept the default location, or specify a different one. A *DataStage* directory is created below the location to contain all DataStage-related directories and files.

Select Suitable Shell

To install or administer DataStage, you must use `sh` or `ksh`.

Set Kernel Parameters

DataStage has certain requirements for the setting of Kernel parameters. When you run the install program, DataStage checks the settings. It warns you if any are too low, and gives you the opportunity of quitting the install so you can reconfigure the Kernel.

Warning Reconfiguration of the UNIX Kernel should only be undertaken by a knowledgeable UNIX user.

The following table gives minimum requirements for a system.

Parameter	Solaris	HP-UX	HP-UX Itanium	AIX	Tru64	LINUX
MSGMAX	8192	32768	65535	N/A	8192	8192
MSGMNB	16384	32768	65535	N/A	16384	16384
MSGSEG	N/A	7168	8192	N/A	N/A	N/A
SEMMNS	111	N/A	4096	N/A	51	128000
SEMMSL	111	N/A	2048	N/A	128	250
SEMMNI	20	1024	4096	N/A	128	1024
SEMOPM	N/A	N/A	N/A	N/A	N/A	32
MAXUPROC	100 per CPU	100 per CPU	4000	100 per CPU	100 per CPU	100 per CPU
NOFILES	1000	1000	65536	1000	1000 * (no. of CPUs)	1000

The following gives guidelines for the minimum settings for the setting of shared memory kernel parameters, these settings apply to all UNIX platforms:

- SHMMAX - maximum shared memory segment size.
Minimum of 536870912. If your system has disk caching turned on and the DISKCACHE variable is set larger than 512, then SHMMAX will need to be larger (see *Shared Memory Disk Caching* for details).
- SHMMNI - shared memory identifiers
The value for this depends on how many client connections, work processes, and IPC links are active at any given time. 2000 is an adequate minimum value for all but the most complex systems. The maximum value for SHMMNI on HP-UX is 1024.
- SHMSEG - maximum attached shared memory segments per process

A minimum setting of 200 should be adequate for this parameter.

These settings give the **generic** names for the kernel parameters. The actual name, and the case, varies for each UNIX platform.

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.

AIX-Specific Kernel Configuration

For Enterprise Edition, we recommend that you set the per-user parameters as follows to ensure that users running Enterprise Edition jobs have sufficient resources.

User Parameter	Setting
fsize (largest file a user may create)	2 GB - 512 bytes (4194303 512-byte blocks)
data (largest data segment – heap – a program can have)	128 MB minimum (262144 512-byte blocks)
stack (largest stack size a program can have)	32 MB minimum (65536 512-byte blocks)
fsize (maximum amount of physical memory a user's process may use)	64 mb minimum (131072 512-byte blocks)

HP-UX- Specific Kernel Parameters

On HP-UX, the maximum size of segments, including data and text segments, should be set as follows:

User Parameter	Setting
maxdsiz	2039480320 (1945 MB)
maxssiz	82837504 (79 MB)
maxtsiz	1073741824 (1024 MB)
rss	64 MB minimum (131072 512-byte blocks)

For Enterprise Edition on an HP-UX platform, you also need to set nproc to (2048+8*MAXUSERS), maxusers to 64 and maxuprc to 3000. Nfile should change from around 1000 to 10000.

On HP-UX Itanium, the maximum size of segments, including data and text segments, should be set as follows:

User Parameter	Setting
maxdsiz	1073741824 (1024 MB)
maxdsiz_64bit	4294967296 (4096 MB)
maxssiz	8388608 (8 MB)
maxssiz_64bit	268435456 (256 MB)
maxtsiz	100663296 (96 MB)
maxtsiz_64bit	1073741824 (1024 MB)

Tru64-Specific Kernel Configuration

On all versions of Tru64, we recommend disabling the following kernel parameter:

- advfsSyncMmapPages

If the operating system has been configured to use Segmented Shared Memory (SSM), the ssm_threshold parameter should be set to a minimum of 16777214. If SSM is not in use on the system, the ssm_threshold value should be set to 0.

For Enterprise Edition, we recommend that you set the per-user parameters as follows to ensure that users running Enterprise Edition jobs have sufficient resources.

User Parameter	Setting
VM_MAXVAS (maximum virtual address space per process)	1048576
max_proc_per_user (number of processes an individual user can run)	1000
per_proc_data_size (per-process allocatable memory; set to value of WM-MAXVAS)	1048576
max_per_proc_data_size (maximum per-process allocatable memory)	1073741824

Pre-Install Special Considerations

This section details special consideration there may be depending on what type of UNIX system you have and what is already installed on it. There are no special considerations for LINUX systems.

Solaris

Ensure that you have the following setting in */etc/system*:

```
shmsys:share_page_table=0
```

Multi-Processor Systems

Before installing or upgrading DataStage on a multi-processor Solaris system 2.7 or 2.8 system, you should install one of the following Solaris patches:

- 106980-19 (Solaris 2.7)
- 108827-26 (Solaris 2.8)

Installing Enterprise Edition

If you are upgrading to include Enterprise Edition, you should install the following patches before upgrade:

- 106327-12 (Solaris 2.7)

- 108434-12 and 108993-37(Solaris 2.8)

Solaris 2.9

Before upgrading to this release on a Solaris 2.9 system, a patched version of `cpio` is required. This is available in the following T-PATCH:

- T113280-03

Compaq Tru64

If you are installing release 7.5.1 of DataStage on a Compaq Tru64, and do not have the Compaq C++ 6.2 compiler installed, you **MUST** install the Compaq C++ Run-Time Library before installing. (

The library is provided as part of a redistribution kit, details of which can be found on the Compaq web site at the following location:

`ftp://ftp.compaq.com/pub/products/C-CXX/Tru64/cxx/cxxredist.htm`

The redistribution kit is supplied in the form of a tar file and can be downloaded from the following address:

`ftp://ftp.compaq.com/pub/products/C-CXX/Tru64/cxx/`

It is recommended that you always download the latest redistribution kit. At the present time this is `CXXREDIST651V37.tar`. Note that DataStage has been built with version 6.3-014 of the library so this is the minimum revision required.

Note If you are installing Enterprise Edition, you require the full compiler, see "[C++ Compilers and Run-Time Libraries](#)" on [page 3-13](#).

AIX

If you do not have either the Visual Age C++ 6 or XL C/C++ v7 compiler (and associated packages) installed, you need to install the Visual Age C++ 6 runtime library. This can be downloaded from:

`http://www-1.ibm.com/support/docview.wss?rs=0&q=x1C.rte&uid=swg24004427&loc=en_US&cs=utf-8&cc=us&lang=en`

Note If you are installing Enterprise Edition, you require the full compiler, see "[C++ Compilers and Run-Time Libraries](#)" on [page 3-13](#).

In order to schedule jobs after installation the permission of `/usr/spool/cron/atjobs` must be changed from 770 to 775 (`rwrxwrx-x`).

NIS

If you are using NIS, you should update the NIS master *etc/services* files to include an entry for dsrpc as port 31538/tcp.

GCI

If you have UniVerse installed on your system and it uses the GCI, you should contact Ascential technical support.

Hardware and Software Requirements

Check that you have the following hardware and software before you install DataStage.

Server

Any computer running UNIX platforms supported by DataStage. The platforms supported by this release are:

- Sun Solaris 2.8, 2.9, 2.10
- IBM AIX 5.1, 5.2, 5.3
- HP-UX 111i(11.11) for PA-RISC
- HP-UX 11iv2(11.23) for Itanium
- HP/Compaq Tru64 5.1, 5.1A, 5.1B
- Red Hat LINUX Advanced Server 3.0
- SUSE LINUX Enterprise Server 9
- USS z/OS 1.3+ (for deployment of parallel jobs from DataStage UNIX server)

If a release is not listed here, it is no longer supported.

Disk Space

You need the following disk space.

On the server:

- 425 MB to install the DataStage Server Edition
- 650 MB to install DataStage Enterprise Edition
- 170 MB for RTI Plug-ins and JVM
- 500 MB temporary storage during install to unpack components

- 3 MB per project to install the DataStage server
- 25 MB of free space in */var*
- Sufficient storage space for any data that is to be held in DataStage tables or files
- Additional space to allow for temporary data storage while a DataStage job is running

On the client:

- 10 MB in the Windows 2000 or Windows XP *system32* directory, as appropriate
- 200 MB in the DataStage install directory for DataStage clients and documentation (typical installation)
- Additional space to allow for temporary data storage while a DataStage job is running

Enterprise Edition

The Enterprise Edition runs on systems that meet the following requirements:

- 200 MB of free disk space for product installation
- 128 - 256 MB or more of memory per processing node, depending on your application
- At least 500 MB of scratch disk space per processing node
- The correct C++ compiler as specified in "[C++ Compilers and Run-Time Libraries](#)."

These are in addition to the other system requirements.

USS Systems

If you intend to run Enterprise Edition jobs on USS systems, the basic requirements of your setup are given in "[Requirements](#)" on page 8-1.

C++ Compilers and Run-Time Libraries

Development Systems

To develop parallel jobs, you need the C++ compiler specific to your platform. You need the C++ runtime library for AIX for all installations (not just Parallel Jobs), details of how to acquire this are given in "[Deployment Systems](#)" below.

The required compilers are:

- For AIX, Visual Age C++ 6.0, XL C/C++ v7
- for Compaq Tru64, Compaq C++ 6.2
- for HP-UX, HP ANSI C++ A.3.56.
- for Sun Solaris Sun Studio 8
- for Red Hat LINUX and SUSE LINUX g++ 3.2.3.

When installing the C++ compiler for your machine, ensure **all** packages are installed.

Deployment Systems

For systems where you only require to run already-compiled parallel jobs, you do not require a full C++ compiler, but you do require certain run-time libraries to be present. The required libraries are:

- For AIX. Depending on the compiler on the development system, either:
 - The VisualAge C++ Version 6 Runtime Libraries for AIX 5.x. The Runtime Library filesets for AIX 5.x can be installed on AIX 5.x using the AIX system administrative tool smit. The libraries are available from the March 2003 PTF, which is accessible at:

http://www-1.ibm.com/support/docview.wss?rs=0&q=x1C.rte&uid=swg24004427&loc=en_US&cs=utf-8&cc=us&lang=en

or:

- The XL C/C++ v7 Runtime Libraries for AIX 5.x. The Runtime Library filesets for AIX 5.x can be installed on AIX 5.x using the AIX system administrative tool smit. The libraries are available from:

http://www-1.ibm.com/support/docview.wss?rs=203&context=SW000&dc=D410&q1=runtime+libraries&uid=swg24007696&loc=en_US&cs=UTF-8&lang=en

- For Compaq Tru64: Compaq C++ run-time library. You need to install this **before** you install DataStage. The library is provided as part of a redistribution kit, details of which can be found on the Compaq web site: <ftp://ftp.compaq.com/pub/products/C-CXX/Tru64/cxx/cxxredist.htm>. The redistribution kit is supplied in the form of a tar file and can be downloaded from: <ftp://ftp.compaq.com/pub/products/C-CXX/tru64/cxx>. We recommend that you always download the latest redistribution kit. At the present time this is CXXREDIST632V20.tar. Note that DataStage has been built with version 6.3-014 of the library so this is the minimum revision required.
- for HP-UX, HP ANSI C++ A.3.50, the runtime library is installed as part of the operating system.

- for Sun Solaris, the runtime library is installed as part of the operating system. No action is required (provided the runtime libraries have not been explicitly removed).
- for Red Hat and SUSE LINUX the the runtime library is installed as part of the operating system. No action is required (provided the runtime libraries have not been explicitly removed).

Client

A PC with an Intel processor (or equivalent) running Windows 2000 service pack 4 or later (Professional, Server, Advanced Server), or Windows XP Professional Service Pack 2. Microsoft Internet Explorer 5.01 SP2 or better. A screen resolution of 800 x 600 or better is recommended with True Color (24-bit). A minimum of 128 MB memory.

You also require .NET framework v1.1.

Note DataStage 7.5.1 server is compatible with DataStage 7.5, 7.5A and 7.5.1 clients but nothing earlier. DataStage 7.5.1 clients may only be used with DataStage 7.5.1 servers.

Networking

The client and server should be networked using TCP/IP network protocol or LAN Manager.

Media

You need a CD-ROM drive that is accessible on the local machine that is your DataStage server.

Installing DataStage Server

To install the DataStage server for the first time on a UNIX platform:

- 1 Log in as *root* if intending to perform a *root* install, or as the user who will be designated product administrator if you want to perform a non-*root* install.
- 2 Insert the DataStage CD-ROM in the drive and, if necessary, mount the drive.

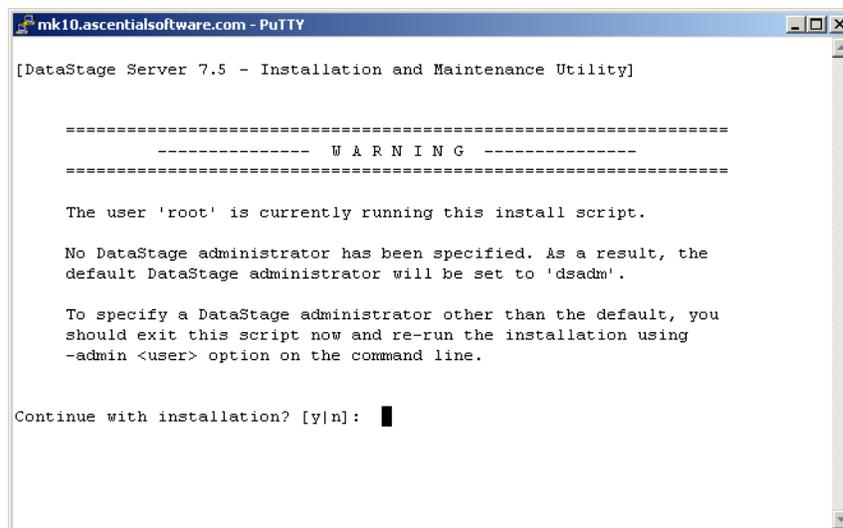
- 3 Switch to the directory where the CD-ROM drive is mounted and run the *Install.sh* program in that directory. Remember that the user running the install must have an appropriate umask setting (022 or 0022 recommended). The suffix or case of the program name may vary slightly according to your platform:

- Solaris – `install.sh`
- AIX – `install.sh`
- HP-UX – ``INSTALL.SH;1``
- Compaq Tru64 – ``INSTALL.SH;1``
- LINUX – ``install.sh``

If you are installing as *root* you can at this point specify the identity of the DataStage Administrative user:

- Solaris – `install.sh -admin userid`
- AIX – `install.sh -admin userid`
- HP-UX – ``INSTALL.SH;1` -admin userid`
- Compaq Tru64 – ``INSTALL.SH;1` -admin userid`
- LINUX – ``install.sh -admin userid``

- 4 A warning screen checks that you are installing a fresh system, not upgrading. Enter **y** to continue.
- 5 The system will tell you whether it is performing a root or non-root installation. It also tells you who the administrative user is. If you are happy with the settings, enter **y**.



```
mk10.ascentialsoftware.com - PuTTY
[DataStage Server 7.5 - Installation and Maintenance Utility]

=====
-----  W A R N I N G  -----
=====

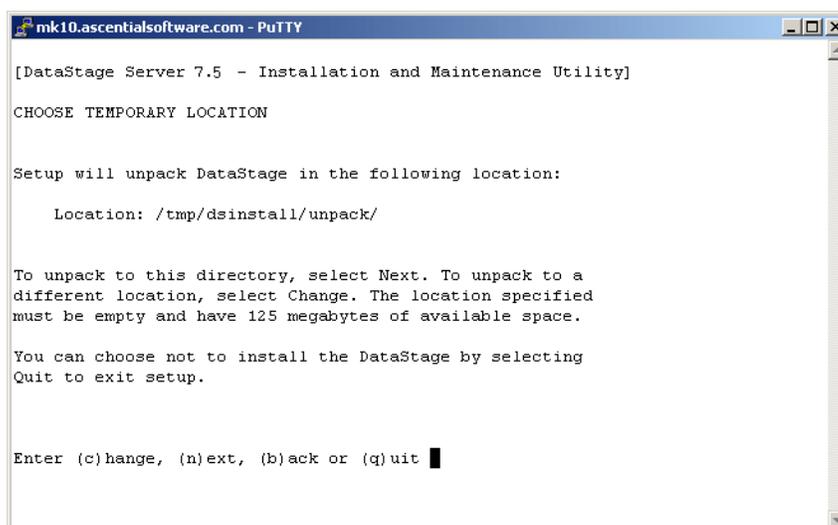
The user 'root' is currently running this install script.

No DataStage administrator has been specified. As a result, the
default DataStage administrator will be set to 'dsadm'.

To specify a DataStage administrator other than the default, you
should exit this script now and re-run the installation using
-admin <user> option on the command line.

Continue with installation? [y|n]: █
```

- 6 If the install detects that you have a version of the UniVerse database installed, it prompts you to specify an alternative port number for your installation of DataStage (by default UniVerse and DataStage use the same port number). Select **(n)ext** to go to the next step.
- 7 The next screen asks you to confirm installation, Select **(n)ext** to go to the next step.
- 8 A screen summarizes the install procedure you have selected. Select **(n)ext** to go to the next step.
- 9 The next screen asks you to acknowledge the license agreement. Enter **y** to accept the agreement.
- 10 As this is a fresh install the next step is the Choose Temporary Location screen. Here you can specify the location where the DataStage server setup files will be unpacked.



```
mk10.ascentialsoftware.com - PuTTY
[DataStage Server 7.5 - Installation and Maintenance Utility]
CHOOSE TEMPORARY LOCATION

Setup will unpack DataStage in the following location:

Location: /tmp/dsinstall/unpack/

To unpack to this directory, select Next. To unpack to a
different location, select Change. The location specified
must be empty and have 125 megabytes of available space.

You can choose not to install the DataStage by selecting
Quit to exit setup.

Enter (c)hange, (n)ext, (b)ack or (q)uit
```

The default directory is */tmp/*. Do one of the following:

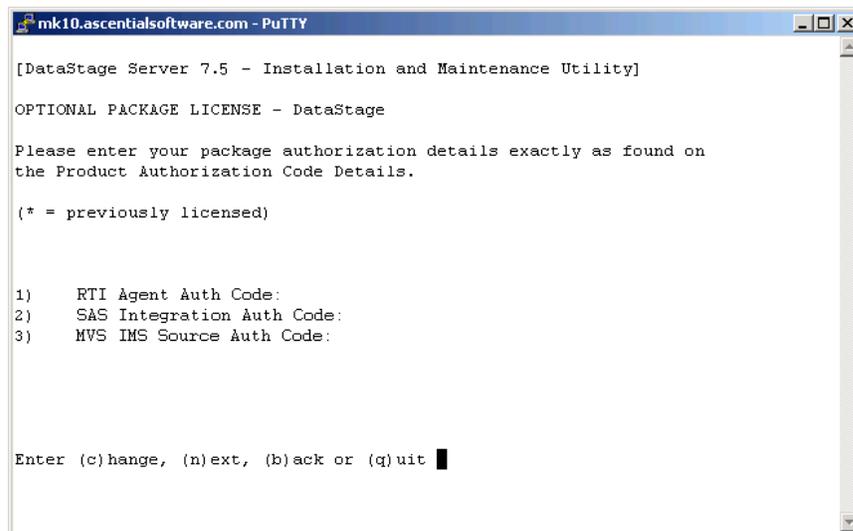
- To change the temporary location, select **(c)hange**, enter the new pathname at the prompt, press **Return**, then select **(n)ext** to go to the DataStage Server License screen.
 - Select **(n)ext** to go to the DataStage Server License screen.
- 11 Answer the following prompts to license the DataStage server:
 - 1) Serial Number:
 - 2) CPU Count:
 - 3) Expiration Date:

 - 4) Enterprise Edition Code:
 - 5) Server Code:
 - 6) Enterprise MVS Edition Code:

Enter each item then press **Return**. You can skip entering any of the codes for versions you are not installing (Enterprise Edition

gives you parallel jobs, Server gives you server jobs, and Enterprise MVS Edition gives you mainframe jobs), but you must enter the code for at least one of these. You are shown the completed license information and offered the chance to amend it. When you are satisfied, select **(n)ext**. This validates the license and displays the License Packages screen.

- 12 Enter the authorization codes for RTI Agent, SAS Integration, MVS IMS Source. If you do not enter codes the packages will not be installed. When you are done, select **(n)ext to go to the** Choose Destination Location screen.

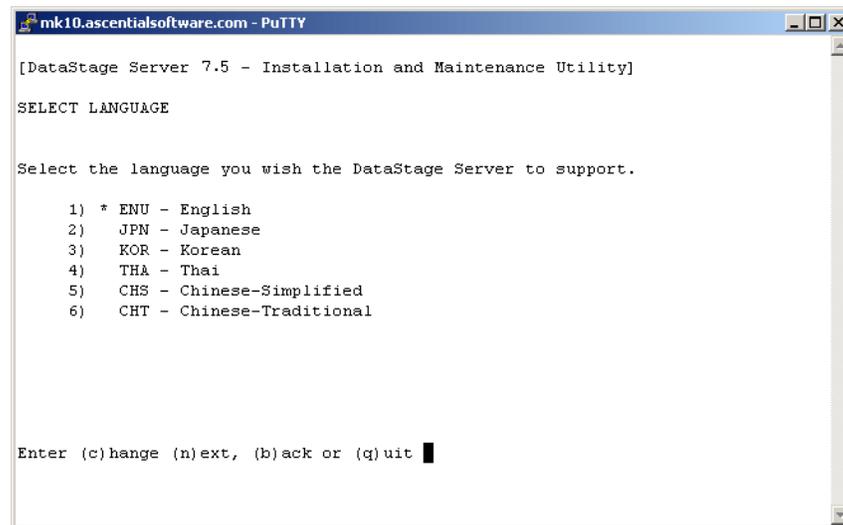


```
mk10.ascentialsoftware.com - PuTTY
[DataStage Server 7.5 - Installation and Maintenance Utility]
OPTIONAL PACKAGE LICENSE - DataStage
Please enter your package authorization details exactly as found on
the Product Authorization Code Details.
(* = previously licensed)
1) RTI Agent Auth Code:
2) SAS Integration Auth Code:
3) MVS IMS Source Auth Code:
Enter (c)hange, (n)ext, (b)ack or (q)uit
```

- 13 Specify the location where you want to install the DataStage server engine or accept the default (see [“Decide on Installation Directory.”](#) on [page 3-7](#)). You can specify a link if required.

To edit the location, select **(c)hange**, enter the new destination and press **Return**.

Select **(n)ext**. The NLS Select Language screen appears (NLS means National Language Support).



```
mk10.ascentialsoftware.com - PuTTY
[DataStage Server 7.5 - Installation and Maintenance Utility]

SELECT LANGUAGE

Select the language you wish the DataStage Server to support.

1) * ENU - English
2) JPN - Japanese
3) KOR - Korean
4) THA - Thai
5) CHS - Chinese-Simplified
6) CHT - Chinese-Traditional

Enter (c)hange (n)ext, (b)ack or (q)uit █
```

- 14** A list of supported languages appears, with the default language selected. Specify the server language to be supported.

To change the default selection, select **(c)hange**, select the appropriate language from the list of available languages by typing its number, and press **Return**.

Select **(n)ext**. The National Language Support screen appears. This screen appears only if the server language selected is English. The default selection is set to **No**. (NLS is installed by default with all other languages.)

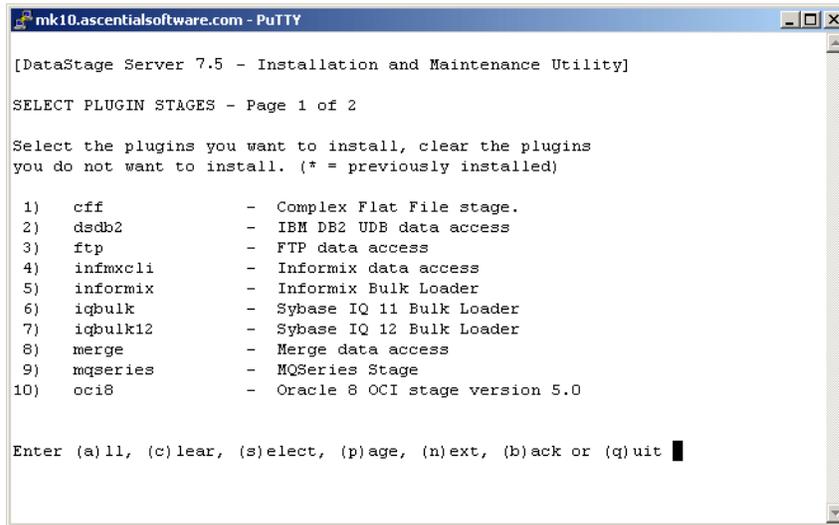
Select **(n)ext**. The Select Plug-in Stages screen appears. Plug-ins are specialized packages that allow you to connect to specific databases or perform other special functions. You can install them at any time, so don't worry if you miss one you need at this stage (information about individual Plug-ins can be found in the *DataStage Packages\PlugIn Stages\Docs\ENU* directory on the install CD-ROM).

- 15** Select **the** Plug-ins you want to install.

Do any of the following:

- Select **(a)ll** to select all Plug-ins.
- Select **(s)elect** select individual Plug-ins.
- Select **(c)lear** to clear selected Plug-ins.
- Select **(p)age** to page through available Plug-ins.

If you choose **select** the screen changes to allow you to choose the Plug-ins you want to install.



```

mk10.ascentialsoftware.com - PuTTY
[DataStage Server 7.5 - Installation and Maintenance Utility]

SELECT PLUGIN STAGES - Page 1 of 2

Select the plugins you want to install, clear the plugins
you do not want to install. (* = previously installed)

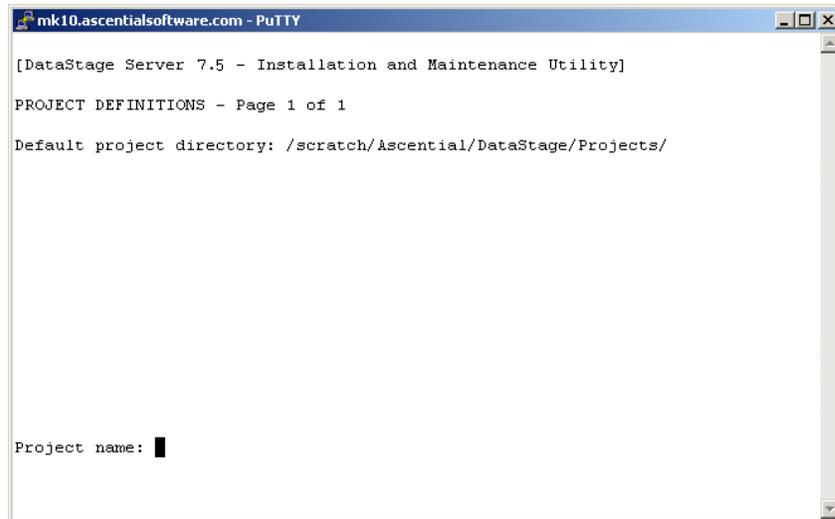
1)  cff          - Complex Flat File stage.
2)  dsdb2       - IBM DB2 UDB data access
3)  ftp         - FTP data access
4)  infmxcli    - Informix data access
5)  informix    - Informix Bulk Loader
6)  iqbulk      - Sybase IQ 11 Bulk Loader
7)  iqbulk12    - Sybase IQ 12 Bulk Loader
8)  merge       - Merge data access
9)  mqseries    - MQSeries Stage
10) oci8        - Oracle 8 OCI stage version 5.0

Enter (a)ll, (c)lear, (s)elect, (p)age, (n)ext, (b)ack or (q)uit █

```

Type in the number of the Plug-in you want to select, a plus sign + appears against it in the list. Type its number again to deselect it. You can select as many Plug-ins as you require.

Select **(n)ext**. The Projects screen appears.



```

mk10.ascentialsoftware.com - PuTTY
[DataStage Server 7.5 - Installation and Maintenance Utility]

PROJECT DEFINITIONS - Page 1 of 1

Default project directory: /scratch/ascential/DataStage/Projects/

Project name: █

```

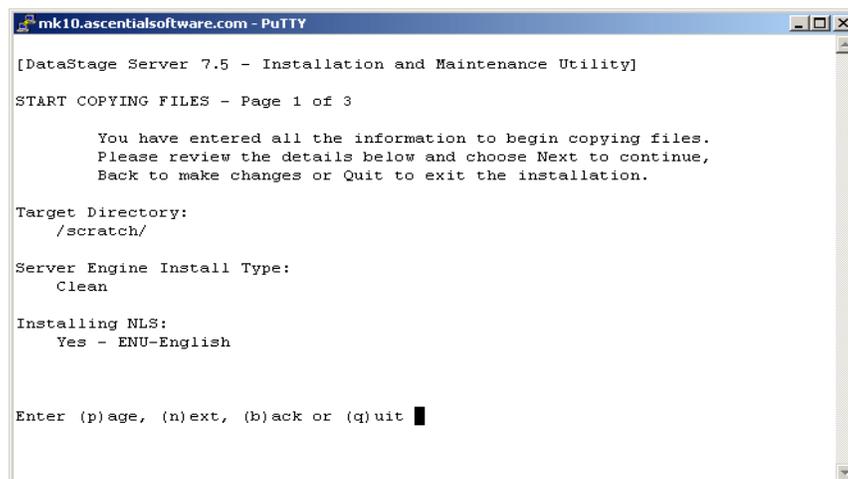
- 16 Specify the names and locations of the projects you want to create. You can create projects in any directory; it does not have to be a subdirectory of the install directory specified in step 13. You might, for example, want to install projects in your users' home directories (e.g., `/users/dwdevs/billG/Projects`). (You can create protected projects at this point if required, see ["Protecting a Project"](#) in *DataStage Administrator Guide* for details about protected projects.)

Once you have entered project details, do any of the following:

- Select **(a)dd** to add additional projects.
- Select **(c)hange** to change project definitions.
- Select **(d)elete** to delete project definitions.
- Select **(p)age** to page through defined projects.

Select **(n)ext**. The Choose Destination screen or the Start Copying Files screen appears, depending on the license you entered (Choose Destination appears for the Enterprise Edition license).

- 17** Select the location for the Parallel engine. Select **(n)ext**. If you have licensed Enterprise Edition, the ORACLE HOME LOCATION - DataStage Enterprise Edition screen appears.
- 18** If you are intending to use Oracle, enter the Oracle home directory, otherwise leave it blank. Select **(n)ext**. The Specify SAS Version for use with DataStage Enterprise Edition screen appears.
- 19** If you are intending to use SAS, enter the version, otherwise leave it set to none. Select **(n)ext**. The DB2 Version screen appears.
- 20** If you are intending to use DB2, enter the version you want to access. Choose between 7, 8, or 'none' ('none' is the default). Select **(n)ext**. The Start Copying Files screen appears.
- 21** The Start Copying Files screen displays information about the location, install type, NLS settings, project definitions, and Plug-in packages. Do one of the following:
 - Select **(p)age** to page through and review the information.
 - Select **(b)ack** to go back and make changes.
 - Select **(n)ext** to exit the setup wizard and start copying files.



```
mk10.ascentialsoftware.com - PuTTY
[DataStage Server 7.5 - Installation and Maintenance Utility]
START COPYING FILES - Page 1 of 3

    You have entered all the information to begin copying files.
    Please review the details below and choose Next to continue,
    Back to make changes or Quit to exit the installation.

Target Directory:
    /scratch/

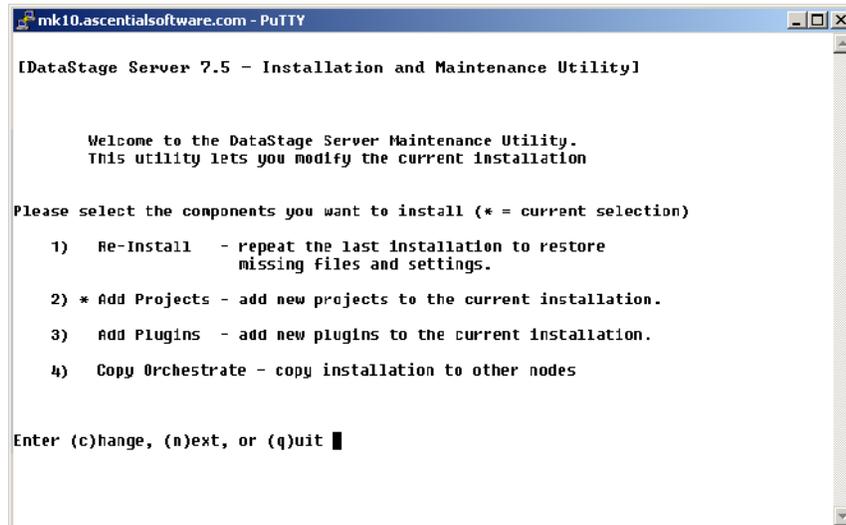
Server Engine Install Type:
    Clean

Installing NLS:
    Yes - ENU-English

Enter (p)age, (n)ext, (b)ack or (q)uit █
```

Reinstalling and Maintenance Menu

To reinstall or access the maintenance menu, follow the instructions for an installation. If the installed version of DataStage matches the CD-ROM version, the setup program reverts to the Maintenance mode screen as shown:



```
mk10.ascentialssoftware.com - PuTTY
[DataStage Server 7.5 - Installation and Maintenance Utility]

Welcome to the DataStage Server Maintenance Utility.
This utility lets you modify the current installation

Please select the components you want to install (* = current selection)

1) Re-Install - repeat the last installation to restore
missing files and settings.

2) * Add Projects - add new projects to the current installation.

3) Add Plugins - add new plugins to the current installation.

4) Copy Orchestrate - copy installation to other nodes

Enter (c)hange, (n)ext, or (q)uit █
```

This allows you to:

- Add new project definitions
 - Add new Plug-ins
 - Copy the parallel engine to other systems in MPP clusters
- 1 Select **(c)hange** to enter change mode. Enter the corresponding number for the option you want (**1, 2, 3, or 4**). The asterisk (*****) indicates which option you are selecting. Press **Return**.
 - 2 Select **(n)ext** and follow the on-screen instructions or select **(q)uit** to quit.

Copying Parallel Engine

Before you can copy the parallel engine to other platforms in a cluster system, you need to ensure that your machine has permission to run *rsh* on the target machines. Then proceed as follows, answering the questions from the copy script:

- 1 The Choose Source Location screen asks you to specify the path of the parallel engine on the current platform (cd to *\$DSHOME/./PXEngine* to find the full path). Enter the path and select **(n)ext**. **The install validates the path then goes onto the next step.**
- 2 The Copy to nodes screen requests a list of nodes to which the parallel engine will be copied. Separate the node names with a space. Select **(n)ext to go to the next step.**

- 3 The Start Copying Files screen displays the copy details and asks for confirmation before it starts copying.

You can also perform this copy function from the UNIX command line if you prefer. See "[Copying the Parallel Engine to Your System Nodes](#)" on [page 6-3](#) for details.

Installation Log Files

Text output generated during the installation process is written to a disk based log file located in the `/tmp/dsinstall/logfiles` directory. Each log file contains information specific to a particular installation instance. Log files are generated for both clean and upgrade installations as well as maintenance mode functions. These log files can be viewed during the installation process using standard UNIX tools, such as `tail` or `more`.

Log file names are of the form `dsinstall.log.MMDDYY.HHMMSS` where `MMDDYY` equals the two digit month, day and year and `HHMMSS` equals the two digit hour, minute and seconds in 24 hour format, for example:

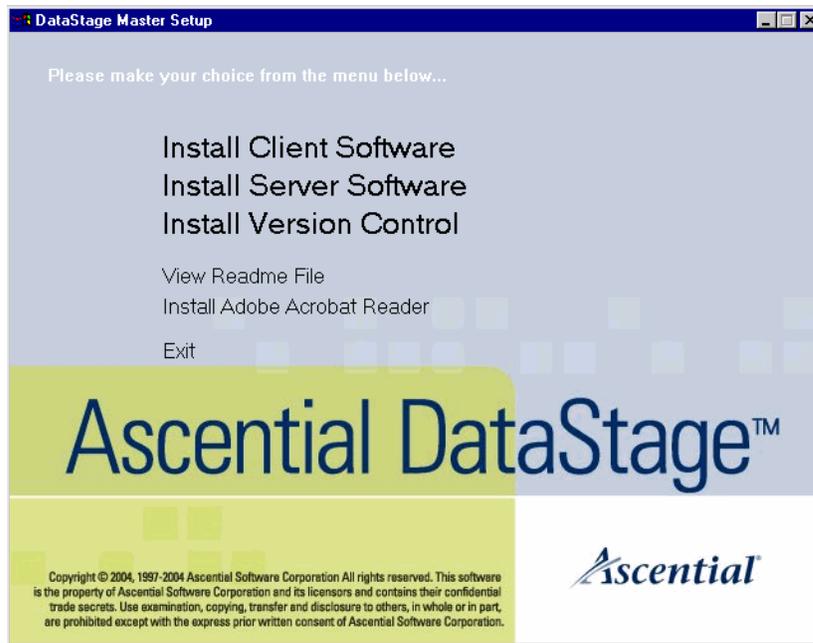
```
dsinstall.log.082201.162431
```

Installing DataStage Clients

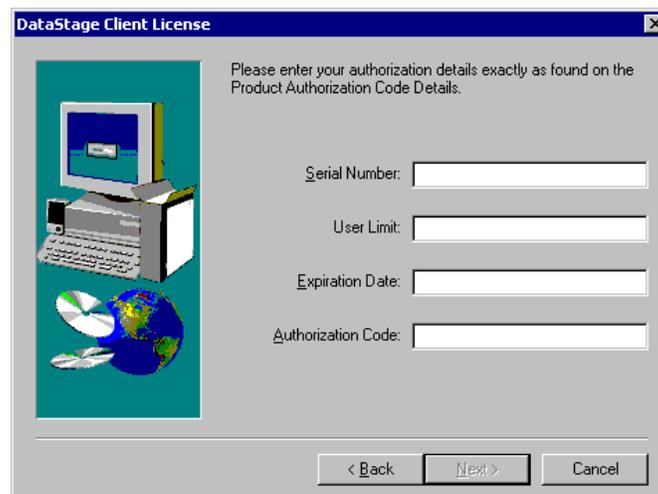
DataStage clients are installed on Windows platforms (Windows XP or Windows 2000). For each platform on which you want to install clients:

- 1 Insert the DataStage client CD-ROM. This contains a master setup program that manages the installation. The DataStage Master Setup screen should appear automatically. If Auto Run is switched

off and the screen does not appear, choose the CD-ROM drive from Windows **Explorer**, then double-click **install.exe**. The DataStage Master Setup screen is shown below.

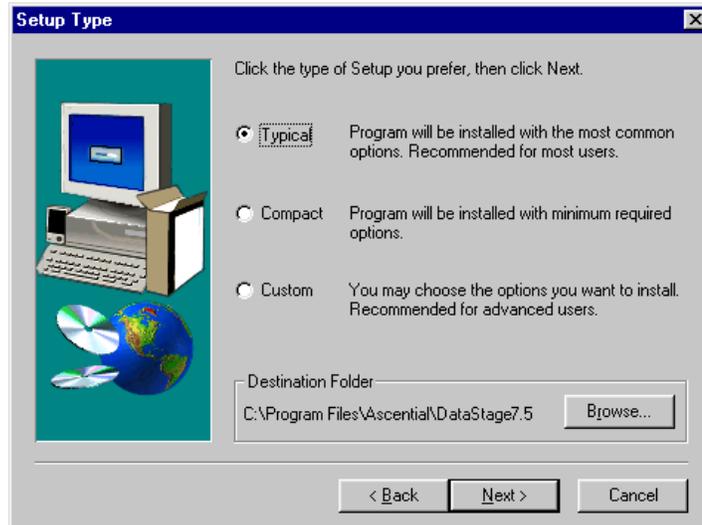


- 2 Choose the **Client** item. A setup program guides you through the install process:
 - a You are asked to enter your authentication details.

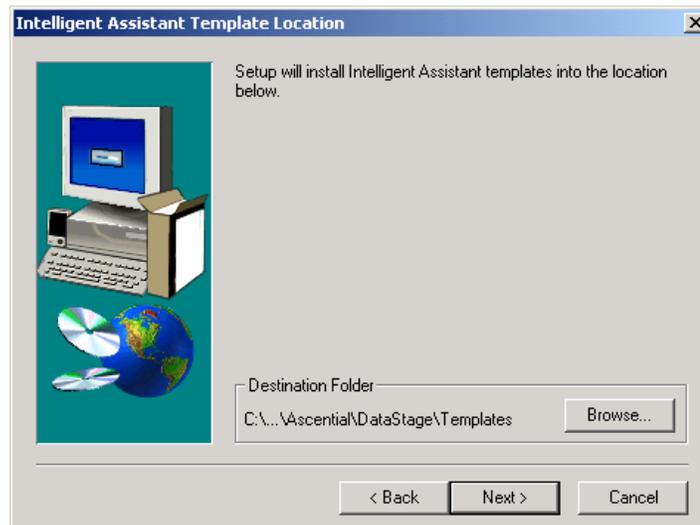


- b You are asked to accept the default installation directory (c:\Program Files\Ascential\DataStage7.5.1) or specify a new one. and to choose between **Typical**, **Compact**, and **Custom** installs (**Typical** is recommended). If you intend to use the DataStage reporting facilities, you need the Documentation Tool. It is installed automatically if you select **Typical**

Installation from the client setup program. You should already have installed the Microsoft ODBC driver manager and the Microsoft Access ODBC driver.



- c A screen asks you where you would like to install the Intelligent Assistant templates. Accept the default destination or browse for a new one.



The client install then proceeds. When it is finished it asks if you want to add shortcuts to your desktop, and whether you want to view the release notes.

- 3 For the latest information on clients, look at the Read Me file.
- 4 In order to use the on-line documentation, you need to install the Adobe Acrobat reader.

Post-Install Checks and Configuration

This section details steps you need to take post-installation to set up database access and ensure smooth and trouble-free operation of DataStage on your UNIX platform.

Non-Root Installs

If you have performed a non-root installation, the following script must be run by the root user in order to enable DataStage:

```
$DSHOME/scripts/DSEservices.sh -add
```

This adds a service entry to `/etc/services` for DataStage. (If you have installed as *root*, this is done automatically).

Tuning the DataStage Server Engine

Due to the various UNIX hardware configurations under which DataStage could be installed, DataStage is shipped with a generically configured DataStage Server Engine (DSEngine) that should be able to be installed under most circumstances. This means, however, that certain (larger) hardware configurations may not take advantage of all available resources, in particular shared memory. A utility called *shmtest* can be used to obtain the best DSEngine tunable values for a given hardware configuration. The *shmtest* utility has the following syntax and options:

```
$DSHOME/bin/shmtest [options]
```

Where possible options are:

- `-c size`. Override default Catalog shared memory segment size of 10485760.
- `-f path`. Specify text file of engine parameters to override current `$DSHOME/.uvconfig` parameters.
- `-h`. Display help details.
- `-n size`. Override default NLS shared memory segment size of 6291456.
- `-s`. Display shmat detail during location determination (for debug use).
- `-v`. Display processing details.

Use *shmtest* to obtain valid shared memory segment attach offset values. With no options specified, the current `$DSHOME/.uvconfig` binary file tunable parameters are used (this requires that the DataStage server engine is running). The `-f` option can be used to specify a text file of engine tunable parameters to be tested (this does

not require the engine to be running). The results of this process can then be used as valid shared memory attach offset values within the `$DSHOME/uvconfig` text file. The new configuration can then be made active.

It is better run the `shmtest` utility against a text file with the DataStage Engine stopped, as the results will then reflect the best offset locations the segments can be attached to. With the DataStage Engine running, the shared memory in use by the running engine will not be available to the `shmtest` utility and offset locations bigger than necessary will be returned in the result.

To use the `shmtest` utility:

- 1 Save your original `$DSHOME/uvconfig` file to `$DSHOME/uvconfig.bak`.

- 2 If the DataStage Engine is running, shut it down using the command:

```
$DSHOME/bin/uv -admin -stop
```

- 3 Execute:

```
$DSHOME/bin/shmtest -f $DSHOME/uvconfig
```

to test the configuration as currently set up in `$DSHOME/uvconfig` text file. When this utility completes, a short display similar to the following will appear:

```
Recommended attach values
Dmemoff  = 0x6fe2000
Pmemoff  = 0x763c000
Cmemoff  = 0x7640000
Nmemoff  = 0x0
```

- 4 Change the tunables `DMEMOFF`, `PMEMOFF` and `CMEMOFF` and `NMEMOFF` in `$DSHOME/uvconfig` to these newly generated values.

Note Do **not** use the values in the example above. Only use those displayed on the screen after running the utility.

- 5 Save the changes to `uvconfig`.

- 6 As `root` or the DataStage administrator, execute:

```
$DSHOME/bin/uv -admin -regen
```

to generate a new `$DSHOME/.uvconfig` binary file.

- 7 Restart the DataStage engine using the command:

```
$DSHOME/bin/uv -admin -start
```

Environment Variables

The DataStage server has a centralized file for storing environment variables called *dsenv*. It resides in *\$DSHOME*, where *\$DSHOME* identifies the DataStage main directory (for example */u1/dsadm/Ascential/DataStage/DSEngine*).

The *dsenv* file is a series of Bourne shell arguments which are referenced during DataStage server startup and can be referenced by interactive users or other programs or scripts. You are likely to want to add new environment variables as you configure DataStage to connect to different databases using Plug-ins or ODBC drivers (see "Configuring Plug-ins" on page 3-30 and "Configuring ODBC Access" on page 3-32).

To emulate the DataStage server environment, in a Bourne shell execute the following command from the *\$DSHOME/DSEngine* directory:

```
. ./dsenv
```

Certain Plug-ins require shared libraries to be loaded and you need to include the library path in an environment variable. The names of the library path environment variables is platform dependent:

Platform	Environment Variable
Solaris	LD_LIBRARY_PATH
HP-UX	SHLIB_PATH
HP_UX Itanium	LD_LIBRARY_PATH
AIX	LIBPATH
Compaq Tru64	LD_LIBRARY_PATH
LINUX	LD_LIBRARY_PATH

Note On HP-UX 11.23 shared libraries are suffixed *.so* instead of *.sl*.

Any environment variable required by the DataStage server should be included in the *dsenv* file. After changing or adding any environment variables, the DataStage server should be stopped and restarted as follows:

- 1 To stop the server:
`$DSHOME/bin/uv -admin -stop`
- 2 To start the server:
`$DSHOME/bin/uv -admin -start`

Ensure that you allow sufficient time between executing stop and start commands (minimum of 30 seconds recommended).

Environment Variables for Enterprise Edition

If you have installed the Enterprise Edition in order to run parallel jobs on your system there are a number of environment variables that you need to set up. This is done in the DataStage Administrator, see "[Setting Environment Variables](#)" in *DataStage Administrator Guide*.

Configuring Enterprise Edition

Full instructions for configuring your platform for use with Enterprise Edition are given in [Chapter 6, "Configuring for Enterprise Edition."](#) Information on setting up the configuration file is given in "[The Parallel Engine Configuration File](#)" in *Parallel Job Developer's Guide* (the DataStage Manager provides facilities for editing the configuration file).

Configuring for Use with PAM

PAM is currently supported on Solaris, HP, AIX, and LINUX platforms. The implementation on each platform is slightly different, and you should consult the documentation supplied with your platform for details. It is assumed you have a working knowledge of PAM and the various authentication modules and strategies.

To implement PAM support on DataStage, you need to:

- Set the AUTHENTICATION flag in the DataStage system file *uvconfig*.
- Add the PAM service entry, *dsepam*, to the PAM configuration file.

DataStage ships with PAM support disabled, and the AUTHENTICATION flag set to 0.

The name and location of the PAM configuration file is platform dependent. For example, versions of PAM of the type generally found on a Solaris system use a PAM configuration file *pam.conf* located in the */etc* directory, and you add entries to this. Versions of Pam of the type generally found on a LINUX system have a specific file for each application located in the */etc/pam.d* directory.

The specific steps you need to add PAM support are:

- 1 Add to or create the PAM configuration file on your platform.
- 2 Stop the DataStage server engine as follows:

```
$DSHOME/bin/uv -admin -stop
```

3 Edit `$DSHOME/uvconfig` and set the AUTHENTICATION flag to 1:

```
# AUTHENTICATION - Specifies the method by which UNIX user
# authentication is done. Currently, the following methods
# are supported:
#
#    0) Standard O/S Authentication (default)
#    1) Pluggable Authentication Module (PAM)
#
# This value should only be changed with a full understanding
# of the implications, as improper setting of this value can
# lead to the environment being unusable.

AUTHENTICATION 1
```

4 Regenerate the DataStage configuration as follows:

```
$DSHOME/bin/uv -admin -regen
```

5 Restart the DataStage server engine as follows:

```
$DSHOME/bin/uv -admin -start
```

Example PAM Configuration Files

On a LINUX system you would need to create a file called `dsepam` in the `/etc/pam.d` directory. This would contain entries similar to the following:

```
##PAM-1.0
auth    required /lib/security/pam_stack.so service=system-auth
passwordrequired /lib/security/pam_stack.so service=system-auth
account required /lib/security/pam_stack.so service=system-auth
```

On a Solaris system you would need to edit the existing `pam.cof` file in the `/etc` directory and add an entry similar to the following:

```
dsepam    auth required    /usr/lib/security/pam_unix.so.1
```

Configuring Plug-ins

DataStage provides a number of Plug-ins which are used to access particular databases and perform other special functions. These are used in server jobs and parallel jobs. The main install program offers you a selection of Plug-ins. If you do not select all the required Plug-ins at initial install you can rerun the install in maintenance mode to select additional ones (see "[Reinstalling and Maintenance Menu](#)" on [page 3-22](#)). You can also download Plug-ins from the Web and install them using the package installer, see "[Installing DataStage Packages](#)" in *DataStage Administrator Guide* for details.

You can determine the available Plug-ins by examining the `packages` directory on the UNIX CD.

Some of the Plug-in technical bulletins still contain references to special files called `/.uvhome` or `/.dshome`. This file determines where

DataStage is installed. From release 7.0 onwards the DataStage installation directory is identified by an environment variable called DSHOME.

The following chart lists the location of the Plug-in package for each platform:

Platform	Pathname
Solaris	/cdrom/packages/solaris
HP-UX	/cdrom/PACKAGES/HP
AIX	/cdrom/packages/aix
Compaq Tru 64	/cdrom/PACKAGES/DEC
LINUX	/cdrom/PACKAGES/ LINUX

Most Plug-ins require you to specify some environment variables in *dsenv*. Up to date information about environment variables are in the technical bulletins supplied with each Plug-in and in *DataStage Plug-in Installation and Configuration Guide*. These are accessible from the main PDF contents list (open this by choosing **Ascential DataStage ► Online Manuals ► DataStage Documentation** from the Start Menu). They are in Acrobat format and require an Acrobat reader.

BCPLoad Plug-In

The following details apply to the BCPLoad Plug-in, which is installed with DataStage by default. The BCPLoad Plug-in enables you to use the bulk copy program (BCP) on SQL Server and Sybase databases.

The BCPLoad stage uses the BCP API in client libraries. The BCPLoad Plug-in does **not** support the following libraries:

- MSDBLIB as a client library on any UNIX platform.
- SYBCTLIB as a client library on an HP-UX platform.
- SYBDBLIB as a client library on a Tru64 platform.

When using a BCPLoad stage with a UNIX server, you must specify SYBCTLIB or SYBDBLIB as client libraries. This is done by setting the **Client Library** property on the Stage editor **Properties** page (see "*BCPLoad Stages*" in *Server Job Developer's Guide*). When using a BCPLoad stage with an HP-UX server, you must specify SYBDBLIB as the client library; when using a BCPLoad stage with a Tru64 server, you must specify SYBCTLIB as the client library. For True64, the library must be the first entry in the LD_LIBRARY_PATH in *dsenv* and _RLD_ARGS must be set to -ignore_unresolved, also in *dsenv*.

Note that Sybase DBLIB is no longer supported on Solaris or AIX platforms, use CTLIB instead.

Configuring ODBC Access

DataStage also allows you to access any database that supports ODBC using database-specific ODBC drivers. DataStage comes complete with a set of ODBC drivers for this purpose which are installed automatically.

ODBC Drivers

Note ODBC driver packs are often updated between major releases of DataStage. You are strongly advised to check your release notes for more up-to-date information about ODBC drivers.

Ascential provides an OEM version of the Data Direct ODBC driver pack with this release. These drivers are licensed solely for use with DataStage and require certain connection parameters to be set in order to function properly. Do not use these drivers with other applications, as licensing errors result.

The ODBC drivers are one of two types, depending on the database being connected to and your platform type:

- non-wire protocol drivers - these require you to install the client software for the database on the DataStage server (the drivers use the API supplied by the database client)
- wire protocol drivers - these do not require database client software (they communicate with the database directly)

The following table lists the the drivers installed with DataStage, which databases they access, and which platforms they are available on. Please see the release notes for information about which ODBC drivers and database versions have been validated at this release of DataStage.

Driver	Database	Platform
DB2 Wire Protocol (VMdb220.so)	DB2 wire protocol UDB 7.1, UDB 7.2, UDB 8.1 DB2 for OS/390 6.1, 7 DB2 for OS/390 and z/OS 7 DB2 for AS/400 V4R5, V5R1, V5R2	Solaris, AIX, LINUX, HP-UX
dBase (VMdbf20.so)	dbase IV, V FoxPro 3.0, 6.0 (3.0 functionality level)	Solaris, AIX, LINUX, HP-UX

Informix Client Driver (VMifc120.so)	Informix OnLine 7.x Informix SE 7.2x Informix Dynamic Server with Universal data Option 9.1.4+ Informix Dynamic Server 9.2x Informix Dynamic Server 9.3x	Solaris, AIX, LINUX, HP-UX
AIX 5.1 Informix Client Driver (VMinf5120.so)	Informix OnLine 7.x Informix SE 7.2x Informix Dynamic Server with Universal data Option 9.1.4+ Informix Dynamic Server 9.2x (this driver requires Informix 2.70UC3 SDK or higher for AIX 5.1 to be installed on the DataStage Server)	AIX 5.1
Informix Wire Protocol (VMinf20.so)	Informix Dynamic Server with Universal Data Option 9.1.4+ Informix Wire Protocol Informix Dynamic Server 9.2x Informix Dynamic Server 9.3x	Solaris, AIX, LINUX, HP-UX
Informix Wire Protocol (VMinf19.so)	Informix Dynamic Server with Universal Data Option 9.1.4+ Informix Wire Protocol Informix Dynamic Server 9.2x Informix Dynamic Server 9.3x	Tru64
Oracle (VMor820.so)	Oracle 7.3.4+, 8.0.5+ Oracle 8i R1, R2, R3 (8.1.5, 8.1.6, 8.1.7) Oracle 9i R1, R2 (9.0.1, 9.2)	Solaris, AIX, LINUX, HP-UX
Oracle Wire Protocol (VM0ra20.so)	Oracle 8i R2, R3 (8.1.5, 8.1.6, 8.1.7) Oracle 9i R1, R2 (9.0.1, 9.2)	Solaris, AIX, LINUX, HP-UX
Oracle Wire Protocol (VM0ra19.so)	Oracle 8i R2, R3 (8.1.5, 8.1.6, 8.1.7) Oracle 9i R1, R2 (9.0.1, 9.2)	Tru64
Progress (VMpro120.so)	PROGRESS 9.1B, 9.1C via SQL92 engine	Solaris, AIX, LINUX, HP-UX
SQLServer Wire Protocol (VMmsss20.so)	MS SQLServer 7.0 MS SQLServer 2000 (with service packs 1, 2 and 3)	Solaris, AIX, LINUX, HP-UX
SQLServer Wire Protocol (VMmsss19.so)	MS SQLServer 7.0 MS SQLServer 2000 (with service packs 1, 2 and 3)	Tru64
Sybase Wire Protocol (VMase20.so)	Sybase Adaptive Server 11.5+ Sybase Adaptive Server Enterprise 12.0 and 12.5	Solaris, AIX, LINUX, HP-UX

Sybase Wire Protocol (VMase19.so)	Sybase Adaptive Server 11.5+ Sybase Adaptive Server Enterprise 12.0 and 12.5	Tru64
Text (VMtxt20.so)	Text files	Solaris, AIX, LINUX, HP-UX

Configuring ODBC Connections

You need to edit three files to set up the required ODBC connections. These are:

- dsenv
- .odbc.ini
- uvodbc.config

All three are located in the *\$DSHOME* directory. (Copies of *uvodbc.config* are also placed in the project directories – see "[uvodbc.config File](#)" on [page 3-36](#).)

Non-wire drivers require different set-up information to wire drivers. Non-wire drivers require information about the location of the database client software, wire drivers require information about the database itself.

For more information on configuring the ODBC environment for your specific DBMS, see the Data Direct Drivers Reference manual *odbceref.pdf* file located in the *\$DSHOME/./branded_odbc/books* directory. You should also check the ODBCREAD.ME file in the *branded_odbc* directory. There is also an html file located in *branded_odbc/odbchelp*.

Once you have configured the required drivers, you should test connectivity as described in "[Testing ODBC Driver Connectivity](#)" on [page 3-38](#).

dsenv File

Any environment variables you need for interactive use of ODBC drivers to make a connection to an ODBC data source must be added to the *dsenv* file. This lets the DataStage server inherit the proper environment for ODBC connections.

For a connection using a wire protocol driver, there are no changes required to *dsenv*.

For a connection using a non-wire protocol driver, you generally need to specify the following in the *dsenv* file:

- environment variables required by the database client software

- database home location
- database library directory

The following gives typical entries for commonly used databases:

```
#Sybase 11 setup
LANG=
Export LANG
SYBASE=/export/home/sybase/sybase;export SYBASE
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$SYBASE/lib:/usr/lib:/lib;export
LD_LIBRARY_PATH
# Oracle 8i
ORACLE_HOME=/space/oracle8i
ORAHOME=/space/oracle8i
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib:$ORACLE_HOME/rdbms/
lib;export LD_LIBRARY_PATH
ORACLE_SID=WSMK5
ORASID=WSMK5
export ORACLE_HOME ORAHOME ORACLE_SID ORASID
#informix XPS 9.3
INFORMIXDIR=/space/informix; export INFORMIXDIR
INFORMIXSERVER=solnew2.1;export INFORMIXSERVER
ONCONFIG=onconfig.solnew2; export ONCONFIG
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$INFORMIXDIR/lib:$INFORMIXDIR/lib/
esql:$INFORMIXDIR/lib/cli;export LD_LIBRARY_PATH
LANG=C
export LANG
#DB2 6.1
DB2DIR=/opt/IBMDB2/V6.1;export DB2DIR
DB2INSTANCE=DB2inst1; export DB2INSTANCE
INSTHOME=/export/home/DB2inst1;export INSTHOME
PATH=$PATH:$INSTHOME/sql1lib/bin:$INSTHOME/sql1lib/adm:$INSTHOME/sql1lib/
misc
export PATH
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$INSTHOME/sql1lib/lib;export
LD_LIBRARY_PATH
THREADS_FLAG=native;export THREADS_FLAG
```

You should stop and restart the DataStage server whenever you edit *dsenv*:

- 1 To stop the server:

```
$DSHOME/bin/uv -admin -stop
```

- 2 To start the server:

```
$DSHOME/bin/uv -admin -start
```

Ensure that you allow sufficient time between executing stop and start commands (minimum of 30 seconds recommended).

***.odbc.ini* file**

The *.odbc.ini* files gives information about connecting to the Database (wire protocol drivers) or the database client (non-wire protocol drivers). If your system uses a mix of drivers, your *.odbc.ini* file will contain a mix of entry types.

Configuration examples for the various platforms are provided in *branded_odbc/Ascential_Tools* directory as follows:

- *aix_4.2.odbc.ini* for AIX systems.
- *sun_4.2.odbc.ini* for SunOS systems.
- *linux_4.2.odbc.ini* for LINUX systems.
- *hpux_4.2.odbc.ini* for HPUX systems.

These files give the minimum parameter settings required to create a working *odbc.ini* file. They each give configurations for the following databases:

- DB2 (UNIX, Windows, AS400)
- Oracle (UNIX)
- Informix (UNIX and Windows)
- MS SQL Server 200 connection
- Sybase (Windows)

***uvodbc.config* File**

This file specifies the DSNs for the databases you are connecting to via ODBC. Entries in *uvodbc.config* have the form:

<name>

DBMSTYPE = ODBC

Name is the ODBC data source name (DSN), this should be the same name as specified in the *.odbc.ini* file. Note that the spaces either side of the equal sign are required.

A typical *uvodbc.config* file is as follows:

```
[ODBC DATA SOURCES]

<localuv>
DBMSTYPE = UNIVERSE
network = TCP/IP
service = uvserver
host = 127.0.0.1
<Sybase1>
DBMSTYPE = ODBC
<Sybase2>
DBMSTYPE = ODBC
<Oracle8>
DBMSTYPE = ODBC
<Informix>
DBMSTYPE = ODBC
<DB2>
DBMSTYPE = ODBC
```

A copy of the *uvodbc.config* file is also placed in each DataStage project directory (for example, */ui/dsadm/Ascential/DataStage/Projects/Dstage1*). 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 the file in *\$DSHOME*. If you alter *uvodbc.config* after creating projects you should copy the edited file to the project directories.

DB2 Connections

Before connecting to a DB2 DSN you need to bind DB2 packages to the DSN. The method differs according to whether you are using a wire protocol driver, or a non-wire protocol driver:

Wire Protocol Driver.

You need to bind a package to every DSN you are going to use to connect to a DB2 database. To create and bind a package to a DSN, from the *\$DSHOME/./branded_odbc/lib* directory enter:

```
bind20 <dsn>
```

Where *<dsn>* is the ODBC data source name. You are prompted for a username and password if one is required.

Third Party ODBC Drivers

You can also use ODBC drivers from third parties for connecting DataStage to different data sources. You might require to connect to a database not supported by the Data Direct driver pack, for example, Redbrick. You should be aware that you can only use one ODBC Driver Manager at a time, however. This means you cannot use third party drivers at the same time as you use the DataStage drivers, and you have to perform some re-configuration to switch between the two types.

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.

Note To use an ODBC driver on an AIX platform other than one of the supplied Data Direct ones, contact Ascential support.

To use a third party ODBC driver:

- 1 Install the ODBC manager and driver(s) on your UNIX platform following the instructions supplied by the driver vendor.
- 2 Configure DataStage to connect to the driver. This involves:

- a Adding required environment variables to the *dsenv* file.
- b Setting up the connection to database client or database in the *.odbc.ini* file (must specify DSN, driver name, and log in information as a minimum).
- c Specify the DSN for connecting to the database in the *uvodbc.config* file.

General information about editing *dsenv*, *.odbc.ini* and *uvodbc.config* are given in "[Configuring ODBC Connections](#)" on [page 3-34](#). Details of the exact information to add should be supplied in the driver vendor literature.

- 3 Include the shared library *libodbc.xx* (the suffix *xx* is either *sl* or *so* depending on platform) for the third party ODBC Manager in the DataStage load library path. Do this by editing *dsenv* and ensuring that the path to the third party Manager precedes that to *\$DSHOME/./branded_odbc/lib* in the shared library path environment variable (see "[Environment Variables](#)" on [page 3-28](#) for details of *dsenv* and the shared library path environment variable).
- 4 Once you have installed and configured your driver, you should test connectivity as described in "[Testing ODBC Driver Connectivity](#)" on [page 3-38](#).

To switch back to using Data Direct drivers you must reverse the procedure in step 3, that is ensure that the DataStage ODBC driver library precedes any third party driver libraries.

Testing ODBC Driver Connectivity

You can test the connectivity of the ODBC connections you have defined for DataStage as follows:

- 1 If you haven't previously done so, *cd* to *\$DSHOME* and set up the DataStage environment by running *dsenv*:

```
./dsenv
```

- 2 Invoke the DataStage engine shell:

```
./bin/dssh
```

The DSEngine shell starts.

- 3 Log to the project:

```
LOGTO project_name
```

Where *project_name* is case sensitive.

- 4 Get a list of available DSNs by typing:

```
DS_CONNECT
```

- 5 Test the required connection by typing:

```
DS_CONNECT DSN
```

Where *DSN* specifies the connection you want to test. Enter username and password as requested, you should then be connected to the specified database.

- 6 Enter `.Q` to quit the connection.

If the connection does not work, consult "[ODBC Connection Problems](#)" on [page 10-3](#) for guidance.

Configuring for NLS Use

If you have installed NLS, you need to edit the `$DSHOME/dsenv` file to specify the locale that the server uses (this is separate from the default locales for projects or jobs, which can be different if required). This enables server jobs to compile correctly. Locate the following section in `dsenv`:

```
#LANG="<langdef>";export LANG
#LC_ALL="<langdef>";export LC_ALL
#LC_CTYPE="<langdef>";export LC_CTYPE
#LC_COLLATE="<langdef>";export LC_COLLATE
#LC_MONETARY="<langdef>";export LC_MONETARY
#LC_NUMERIC="<langdef>";export LC_NUMERIC
#LC_TIME="<langdef>";export LC_TIME
#LC_MESSAGES="<langdef>"; export LC_MESSAGES
```

Take the following steps:

- 1 Replace all occurrences of `<langdef>` with the locale used by the server (the locale must be one of those listed when you use the `locale -a` command).
- 2 Remove the `#s` at the start of the lines.
- 3 Stop and restart the DataStage server:

To stop the server:

```
$DSHOME/bin/uv -admin -stop
```

To start the server:

```
$DSHOME/bin/uv -admin -start
```

Ensure that you allow sufficient time between executing stop and start commands (minimum of 30 seconds recommended).

Upgrading Non-Root Installations

If you installed DataStage using an id other than *root* (see "[Types of Install](#)" on [page 3-1](#)), you can subsequently upgrade DataStage to have one or more of the features of a *root*-installed system. You do

this by running a set of scripts. These scripts must be run by *root*, but your system administrator can examine the scripts and assure themselves that they do not compromise security in any way.

The features that you might want to enable in this way are:

- Auto startup (DataStage services restart automatically when the machine restarts).
- Impersonation mode (DataStage server will run with the id and permissions of the user who logged in).

The scripts used to enable or disable these features are:

- *DSEnable_autostartup.sh*. Enables dsengine auto startup at system boot time.
- *DSEdisable_autostartup.sh*. Disables dsengine auto startup at system boot time.
- *DSEnable_impersonation.sh*. Enables impersonation features.
- *DSEdisable_impersonation.sh*. Disables impersonation features.
- *DSEsetperms.sh*. Called by *DSEnable_impersonation.sh* to set permissions.
- *DSEresetperms.sh*. Called by *DSEdisable_impersonation.sh* to reset permissions.
- *DSEservices.sh*. Used to add or update dsrpc entry in */etc/services* file. This must be run on non-root installed systems to enable DataStage (see "[Post-Install Checks and Configuration](#)" on [page 3-26](#)).

You can establish the current operational state of the system by issuing the command:

```
$DSHOME/bin/uv -admin -info
```

The scripts are located in *\$DSHOME/scripts*.

You can also enable or disable the auto start and impersonation features using the `uv -admin` command as follows:

- `uv - admin -autostart on`. Enable auto start.
- `uv - admin -autostart off`. Disable auto start.
- `uv - admin -impersonation on`. Enable impersonation.
- `uv - admin -impersonation off`. Disable impersonation.

4

Upgrading Windows Systems

This chapter describes the process of installing DataStage Release 7.5.1 onto a Windows system with a previous version of DataStage.

You will need to upgrade all server and client systems in your DataStage configuration.

If you are upgrading from releases earlier than 7.5, you should check [Appendix A](#) for issues about your particular upgrade.

Upgrade Issues

Job Executables

When you upgrade from releases earlier than 7.0, all your current DataStage job executable files will be removed. This is to ensure that you recompile before running a job in Release 7.5.1. DataStage now has a job compiler tool which allows you to compile or generate multiple jobs without opening them in the DataStage Designer. See "[Compiler Wizard](#)" in *DataStage Designer Guide* for details.

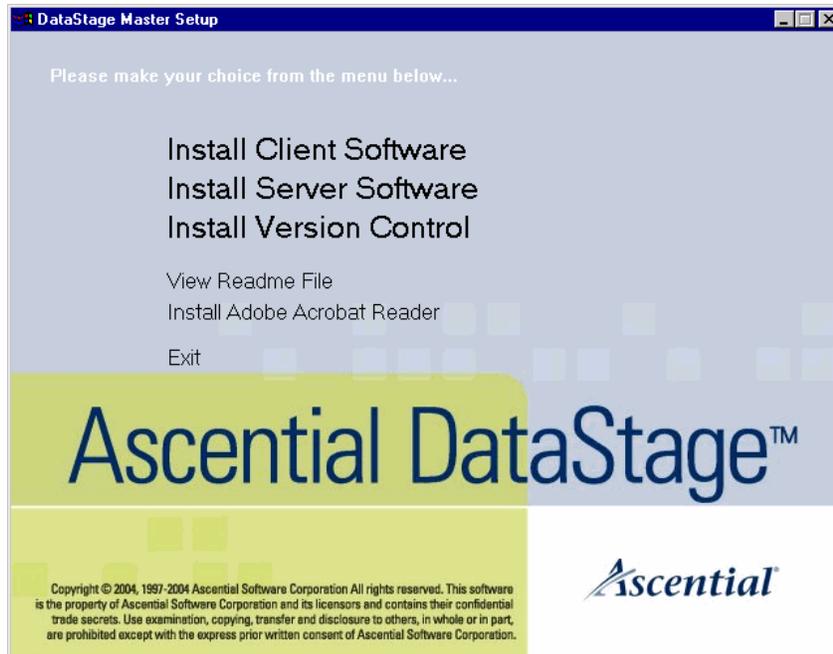
XML Pack

If you are upgrading from release 7.0 or 7.1, you must uninstall your current XML Pack client before upgrading DataStage. To do this:

- 1 Got to Windows control panel.
- 2 Choose Add or remove programs.
- 3 Select DataStage XML Pack and click **Remove**.

The Upgrade Process

The DataStage CD-ROM contains a master setup program that manages the installation. To view the DataStage Master Setup screen, insert the CD-ROM in the drive. The screen should appear automatically. If Auto Run is switched off and the screen does not appear, choose the CD-ROM drive from Windows Explorer, then double-click **install.exe**. The DataStage Master Setup screen is shown below.

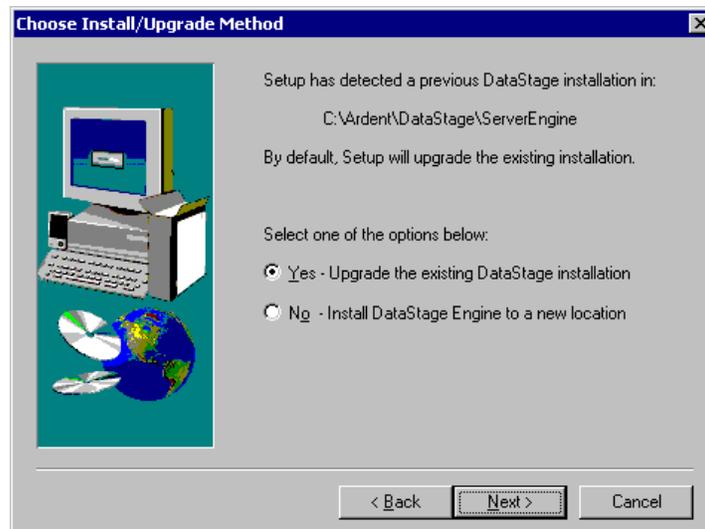


From this screen you can upgrade both DataStage client software and DataStage server software. Just choose the Client or Server options and a setup program guides you through the process. You can also install or upgrade DataStage Version Control (See *Version Control Guide*).

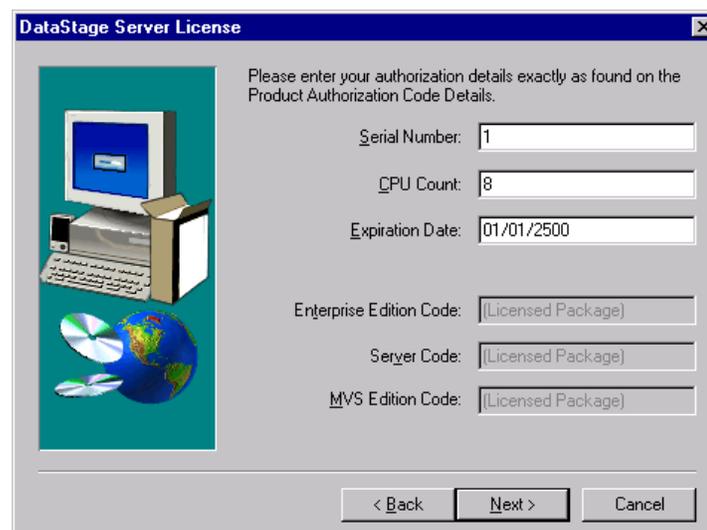
Server Upgrade

The server upgrade process is as follows:

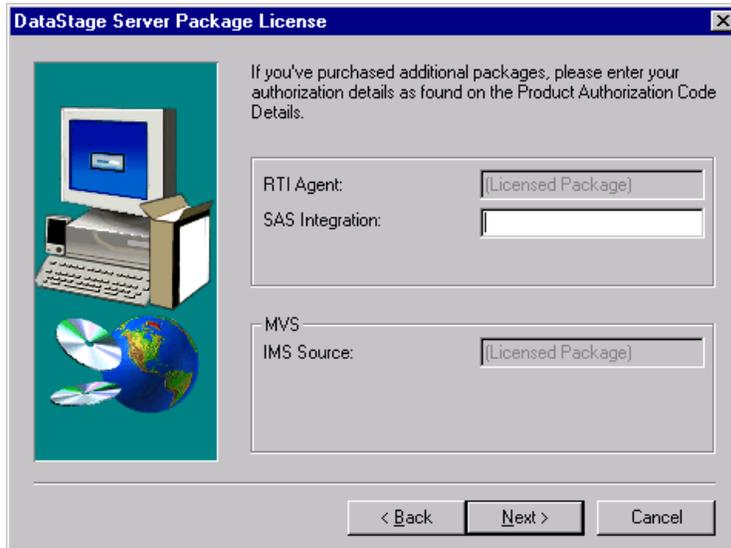
- 1 If you are upgrading from release 4.x or earlier, the install asks you whether you want to upgrade the installed version or preserve the underlying UniVerse as a coresident system (see "[Coresident UniVerse Systems – Windows Systems](#)" on page A-1).



- 2 You are asked to enter your authentication details. On an upgrade the serial number, CPU count, and expiration date fields are filled in automatically, as are the authorization codes for the DataStage edition or editions you are upgrading. You can also add a new edition here if required. You can edit the other fields if, for example, you are upgrading to a license with more CPUs.



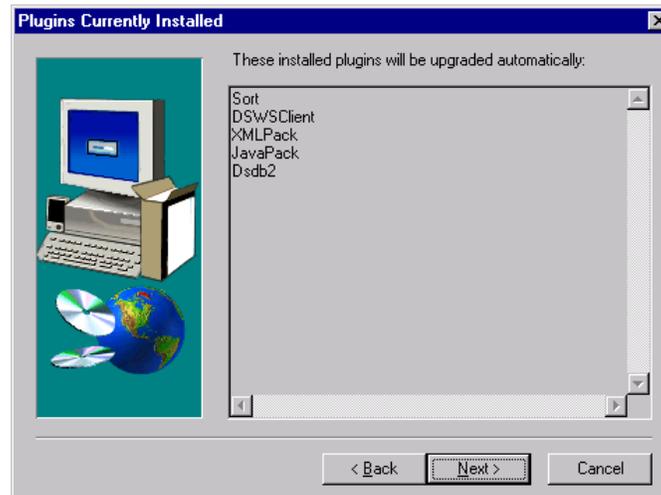
- 3 If you have previously installed any of the optional packages, these are automatically upgraded. Otherwise you are given the opportunity to install them now by inputting authorization codes for them.



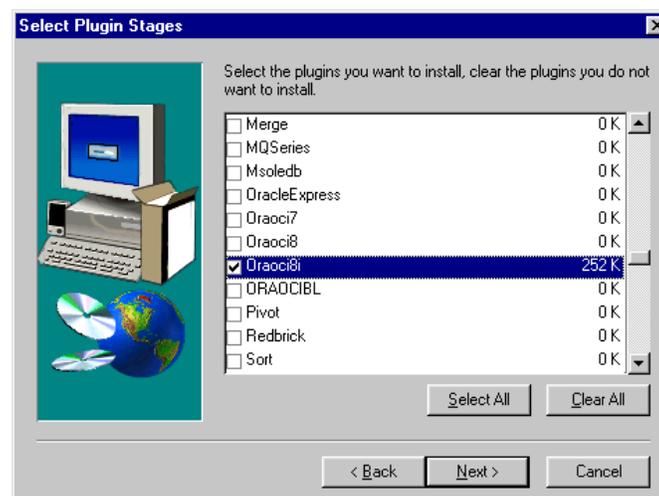
- 4 If your previous version of DataStage did not have NLS installed, you are asked if you want to install it now.



- 5 You are shown what Plug-ins you currently have installed, and which will be upgraded automatically:



- 6 You are asked to select new Plug-ins you want to install. Any Plug-ins already installed will be upgraded automatically (unless they were originally installed with the package installer, in which case they will appear in the list of Plug-ins and should be selected in order to be upgraded).

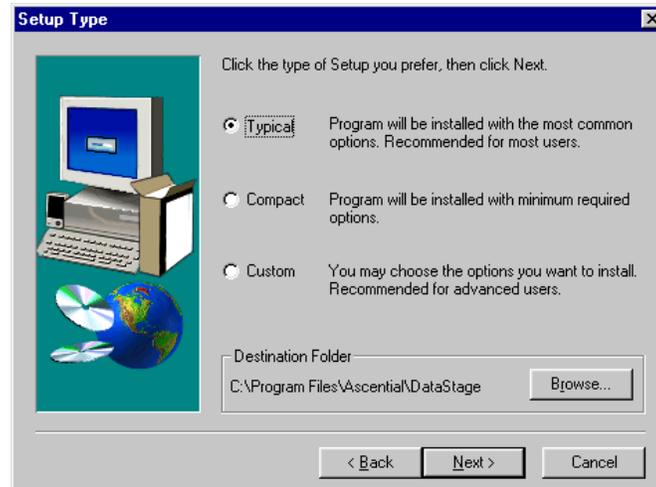


- 7 You are asked if you want to add any more projects.

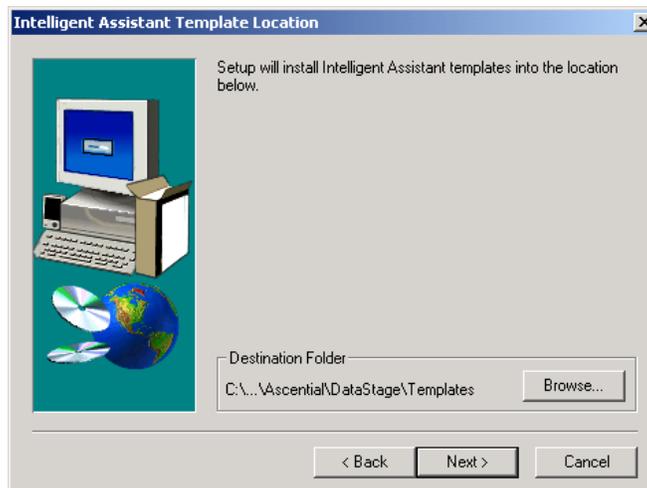
The server upgrade then proceeds. When it finishes it asks if you want to view the release notes and/or start the DataStage Services (you need to start the services before you can run DataStage).

If the version of server software that you are installing is already installed, the setup program enters maintenance mode, from where you can repeat the previous install, or add new projects or Plug-ins.

chosen to overwrite the existing DataStage client, you should not change the **Destination Folder**, if you have chosen to preserve the existing client then you can accept the default destination for the new install (C:\Program Files\Ascential\DataStage7.5.1) or browse for a new destination.



- 4 A screen asks you where you would like to install the Intelligent Assistant templates. Accept the default destination or browse for a new one.



The client install then proceeds. When it is finished it asks if you want to add shortcuts to your desktop, and whether you want to view the release notes.

Note If you want to install the Multi-Client Manager (which allows you to switch between different versions of DataStage client on the same machine), you should select the **Custom** installation mode and explicitly select the Multi-Client Manager.

After installing the client and server software, check that the DataStage configuration meets your requirements. For information about configuring DataStage, see *DataStage Administrator Guide*.

5

Upgrading UNIX Systems

This chapter describes the process of installing DataStage Release 7.5.1 onto a UNIX system with a previous version of DataStage. If you are upgrading from releases earlier than 7.5, you should check [Appendix A](#) for issues about your particular upgrade.

You need to upgrade the DataStage Server on your UNIX platform, and upgrade all the clients you have installed on Windows platforms (unless you intend to maintain earlier clients in order to connect to other servers – in which case you will need to install and use the Multi Client Manager, see [page 5-18](#)).

You should check the Read Me file and "[Problems and Restrictions](#)" on [page 10-4](#) before upgrading to check for any platform-specific issues that might affect your upgrade. The read me file can be found in the top level CD-ROM directory and is called *readme.htm*.

If you are upgrading to include Enterprise Edition, be sure to read [Chapter 6, "Configuring for Enterprise Edition,"](#)

You should be familiar with UNIX system administration in order to perform an upgrade.

Types of Upgrade

If you are upgrading from version 7.0 or later, here are two basic types of upgrade on a UNIX system: root and non-root. In either mode, you can further decide whether you are going to preserve the existing DataStage and whether to install multiple instances of the new server.

If you are upgrading from a version earlier than 7.0, you must perform the upgrade as root. You can still preserve the earlier version of

DataStage and/or install multiple instances of the new DataStage server.

Root Upgrade

DataStage upgraded by *root* is installed in impersonation mode with auto start enabled. During upgrade an administrative user is specified who is authorized to carry out day-to-day administrative tasks. By default this is *dsadm*, but you can specify another user if required.

In impersonation mode, the DataStage server inherits the identification and permissions of the user who has logged onto it. The DataStage server will then execute all functionality using that user's identification and permissions.

Auto start is enabled with impersonation mode, so that the DataStage services start automatically whenever the machine is restarted.

Non-Root Upgrade

If your original DataStage was installed by a non-root user, you can choose to perform the upgrade as this user. DataStage will be installed in non-impersonation mode and have auto start disabled.

Under non-impersonation mode, the DataStage server inherits the identification and permissions of the user who started the DataStage services (usually the product administrator). The DataStage server then executes all functionality using this identification and associated permissions, regardless of the actual id of the user logged in.

Because the server runs with the permissions of the administrative user, every user who logs in will be able to perform administrative tasks.

With this version the administrator will need to manually start the DataStage services whenever the machine is restarted; auto start is not available.

It is intended that after a non-*root* upgrade certain steps are taken by *root* to render DataStage fully functional, in particular:

- DataStage must be enabled by the *root* user running the script `$DSHOME/scripts/DSEservices.sh -add`. This adds a service entry to `/etc/services` for DataStage.
- Root must take steps to enable impersonation mode – otherwise no one can log in via a DataStage client. This can be done by running the script `$DSHOME/scripts/DSEenable_impersonation.sh`.

A number of scripts are provided which can be run by *root* in order to uprate a non-root installed system to give more functionality. Your system administrator can examine these scripts and assure themselves of their security before uprating the system, see "[Uprating Non-Root Installations](#)" on page 5-30.

A typical scenario for using non-root upgrade would be as follows:

- 1 Perform the non-*root* upgrade.
- 2 Have *root* enable DataStage by running *DSEservices.sh -add* (this can be run remotely).
- 3 Test basic DataStage functionality on your system by using *dsjob* to run sample jobs from the command line. (*Dsjob* is described in "[DataStage Development Kit \(Job Control Interfaces\)](#)" in *Server Job Developer's Guide* and "[DataStage Development Kit \(Job Control Interfaces\)](#)" in *Parallel Job Advanced Developer's Guide*.)
- 4 Have root uprate the system for impersonation mode and auto-startup by running the scripts described in "[Uprating Non-Root Installations](#)" on page 5-30. Again, these can be run remotely.

Upgrade Issues

Job Executables

When you upgrade from releases earlier than 7.0, all your current DataStage job executable files will be removed. This is to ensure that you recompile before running a job in Release 7.5.1. DataStage now has a job compiler tool which allows you to compile or generate multiple jobs without opening them in the DataStage Designer. See "[Compiler Wizard](#)" in *DataStage Designer Guide* for details.

XML Pack

If you are upgrading from release 7.0 or 7.1, you must uninstall your current XML Pack client before upgrading DataStage. To do this:

- 1 Got to Windows control panel.
- 2 Choose Add or remove programs.
- 3 Select DataStage XML Pack and click **Remove**.

UNIX Upgrade Checklist

This section provides a checklist of the steps you should take when upgrading DataStage:

- 1** Check that your platform is supported by DataStage Release 7.5.1, and upgrade the platform if required (see ["Platforms Supported by DataStage Release 7.5.1"](#) on page 5-5).
- 2** If upgrading from Release 4.x or earlier, you need to:
 - a** Decide whether you want to maintain a UniVerse database on your system as a coresident system (see ["Coresident Systems – UNIX Systems"](#) on page A-2).
 - b** Set up the special *dsadm* user (this enables you to perform administration tasks without logging in as *root*). You should also review the setup for your existing DataStage users as they should all belong to the same primary group as *dsadm* (see ["Adding the dsadm User"](#) on page 5-5).
- 3** You need to decide whether you want to preserve your existing ODBC drivers and configuration, or change to using the drivers supplied with DataStage 7.5.1 (see ["ODBC Configurations"](#) on page 5-6).
- 4** Back up files that preserve configuration of your existing DataStage system (specifically back up the \$DSHOME/*dsenv* file).
- 5** Mount the installation CD and run the DataStage install script. Verify install is successful.
- 6** Upgrade DataStage clients on Windows platforms.
- 7** Perform post-upgrade configuration:
 - a** If you wish to use authentication other than standard UNIX authentication, configure DataStage to use PAM (see ["Configuring for Use with PAM"](#) on page 5-19).
 - b** Perform configuration required for any new Plug-in stages you have installed with DataStage upgrade (see ["Configuring Plug-ins"](#) on page 5-22).
 - c** If you upgraded to include Enterprise Edition, configure your system in accordance with the instructions given in [Chapter 6](#).
 - d** Restore the configuration if you have retained your old ODBC drivers, or reconfigure if you have installed the new drivers (see ["Configuring ODBC Access"](#) on page 5-23).
 - e** Verify coresident installs (see ["Upgrading Non-Root Installations"](#) on page 5-30).

If you encounter any problems after your upgrade, see [Chapter 10](#), ["Troubleshooting."](#)

Pre-Upgrade Checks and Set Up

This section details set up steps you have to perform and configuration adjustments you may have to make before upgrading the DataStage server on your UNIX platform.

Platforms Supported by DataStage Release 7.5.1

The platforms supported by this release are:

- Sun Solaris 2.8, 2.9, 2.10
- IBM AIX 5.1, 5.2, 5.3
- HP-UX 111i(11.11) for PA-RISC
- HP-UX 11iv2(11.23) for Itanium
- HP/Compaq Tru64 5.1, 5.1A, 5.1B
- Red Hat LINUX Advanced Server 3.0
- SUSE LINUX Enterprise Server 9
- USS z/OS 1.3+ (for deployment of parallel jobs from DataStage UNIX server)

Note in particular that DataStage 7.5.1 does not support Solaris 2.6. If you are currently running DataStage on Solaris 2.6 you need to upgrade to a newer version of Solaris before upgrading DataStage. We recommend that you take the following steps:

- 1 Export all DataStage projects to .dsx or .xml file(s) and backup all critical data.
- 2 Upgrade Solaris to Solaris 2.7 or 2.8.
- 3 Upgrade DataStage to DataStage 7.5.1.

Select Suitable Shell

To install or administer DataStage, you must use *sh* or *ksh*.

Adding the *dsadm* User

From Release 5.1 onwards DataStage administration on a UNIX platform can be carried out by the special administration user, *dsadm*. If you are upgrading from Release 4.x or earlier, you will need add this user. *dsadm* must be set up with a primary group which is the same as the primary group of existing DataStage users. Throughout this documentation `<dsadmhome>` refers to the *dsadm* user's home directory.

If you are upgrading from release 7.0, 7.1, or 7.5, your administration user may have a different id to *dsadm*. For all descriptions, substitute this user id for *dsadm*.

Note On Compaq Tru64 systems, the *dsadm* user must be a member of the system group (group number 0). This is necessary to ensure that *dsadm* can administer DataStage correctly.

ODBC Configurations

DataStage 7.5.1 comes complete with a set of Data Direct ODBC drivers. These support client and Wire Protocol connections for most common databases on most platforms. Wire protocol drivers can connect directly to the target database and do not require the client software for the database to be installed on your UNIX platform. (A list of driver type versus platform and database type is given in "[Configuring ODBC Access](#)" on [page 5-23](#)).

If you are using ODBC drivers, and choose to upgrade to the new Data Direct drivers, you will have to perform some reconfiguration post-install.

If you want to retain the existing ODBC drivers, you must take steps to preserve the set up files **before** you upgrade. Specifically you need to:

- 1 Back up the `$DSHOME/dsenv` file.
- 2 Back up the `$DSHOME/.odbc.ini` file
- 3 Back up the `$DSHOME/uvodbc.config` file
- 4 Back up the `uvodbc.config` file in each of the project directories.

(The upgrade process backs up `dsenv` and `uvodbc.config` automatically, but we recommend taking your own back-ups too).

The existing `branded_odbc` directory is renamed `branded_odbc.xxx` where `xxx` is the previous `odbc` release number.

If you are upgrading from a version of DataStage earlier than Release 4.x DataStage may no longer support your old ODBC drivers, so we recommend that you upgrade to Data Direct drivers supplied with this release.

Back Up Files

We strongly recommend that you back up existing DataStage projects before upgrading DataStage. Use the DataStage Manager client to export projects to a `.dsx` or `.xml` file. These can then be subsequently

reimported into DataStage if there are any problems. (See *"Importing, Exporting, and Packaging Jobs"* in *DataStage Manager Guide* for more details about exporting and importing projects.)

We also recommend that you back up your `$DSHOME/dsenv` file, as this contains environment variables important for the operation of Plug-ins and ODBC connections. If you use ODBC connections, see also *"ODBC Configurations"* on page 5-6. (The upgrade process backs up `dsenv` automatically, but we recommend taking your own back-ups too.)

System Requirements

Ensure you have at least 25 MB free space in `/var`.

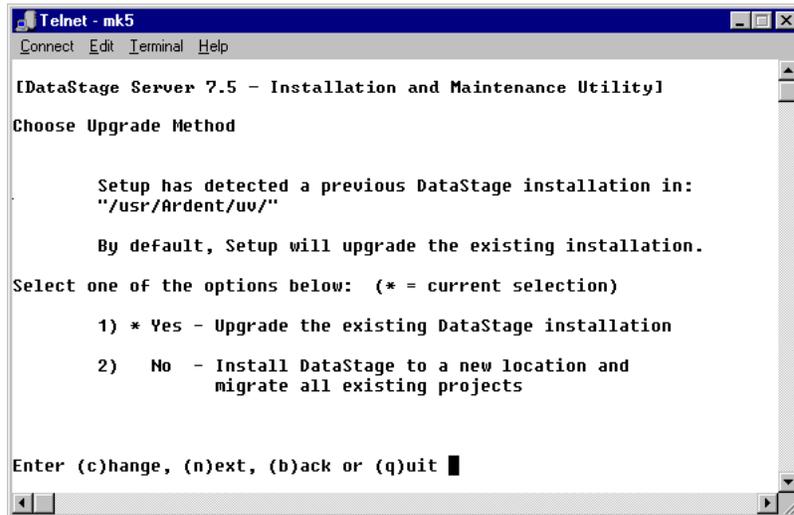
Upgrading DataStage

To upgrade an existing DataStage server on a UNIX platform:

- 1 If you are upgrading from a version earlier than release 7.0, log in as `root`. If you are upgrading from version 7.0 or later, and do not wish to upgrade as root, log in as the designated product administrator user (you must have previously installed the product as the designated product administrator user).
- 2 Insert the DataStage CD-ROM in the drive and, if necessary, mount the drive.
- 3 Switch to the directory where the CD-ROM drive is mounted and run the `install.sh` program in that directory. The suffix or case of the program name may vary slightly according to your platform:
 - Solaris – `install.sh`
 - AIX – `install.sh`
 - HP-UX – ``INSTALL.SH;1``
 - Compaq Tru64 – ``INSTALL.SH;1``
 - LINUX – ``install.sh``
- 4 Select **(n)ext** on the Welcome screen to go to the next step.

If you are upgrading from DataStage Release 4.x or earlier, the install will detect that you have a previous version of DataStage on your system and display the Choose Upgrade Method screen. This allows you to specify whether you want to upgrade the existing DataStage (the default option) or preserve it as a coresident UniVerse installation and install the new DataStage to a

different location. Select **c(hange)** if you want to switch options. When you are happy with your selection, select **(n)ext**.



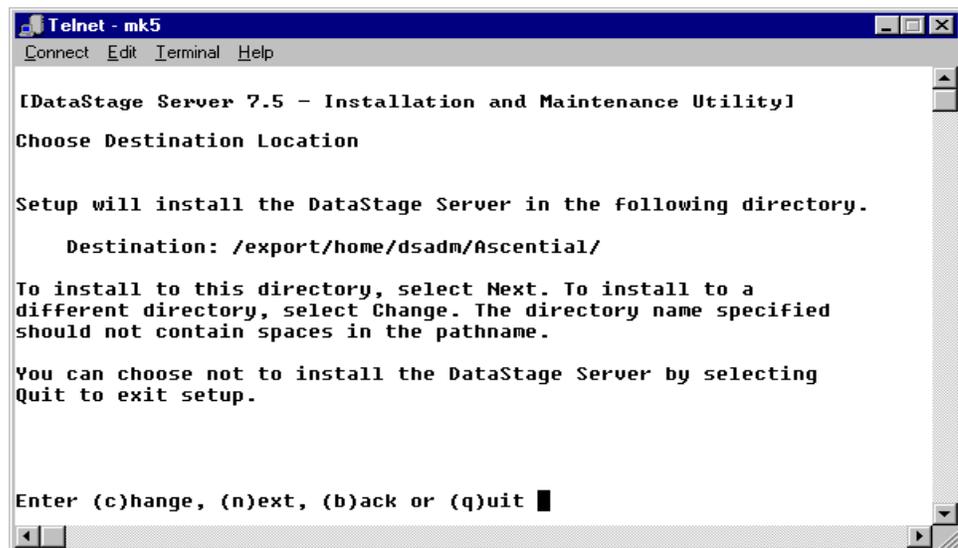
If the install detects no previous versions, you are warned that the installation will be treated as a new one, and given the opportunity to quit. (In this case you should set and export the DSHOME environment variable to indicate the location of the existing DataStage server engine, or explicitly specify the location on the command line when you rerun the install by using the option `-home pathname`. In either case this should allow the install to detect that this is an upgrade.)

- 5 The Choose Temporary Location screen appears. Here you can specify the location where the DataStage server setup files will be unpacked. The default directory is `/tmp/`.

Do one of the following:

- To change the temporary location, select **(c)hange**, enter the new pathname at the prompt, press **Return**, then select **(n)ext** to go to the DataStage Server License screen.
 - Select **(n)ext** to go to the DataStage Server License screen.
- 6 As this is an upgrade, Serial Number, CPU Count, and Expiration Date are already populated. You can alter these if required if you are upgrading with a new license. Otherwise you need only specify the Authorization Code for Server (at this upgrade this will always be licensed).
 - 7 If you already have XE/390 (i.e., Enterprise MVS Edition) or DataStage Parallel Extender (i.e., Enterprise Edition) installed, these will be upgraded automatically. Otherwise you can install

them now by entering license codes for them. If you do not enter codes the editions will not be installed. Choose **n(ext)** to validate the license and go to the Choose Destination Location screen.



```
Telnet - mk5
Connect Edit Terminal Help

DataStage Server 7.5 - Installation and Maintenance Utility
Choose Destination Location

Setup will install the DataStage Server in the following directory.

Destination: /export/home/dsadm/Ascential/

To install to this directory, select Next. To install to a
different directory, select Change. The directory name specified
should not contain spaces in the pathname.

You can choose not to install the DataStage Server by selecting
Quit to exit setup.

Enter (c)hange, (n)ext, (b)ack or (q)uit █
```

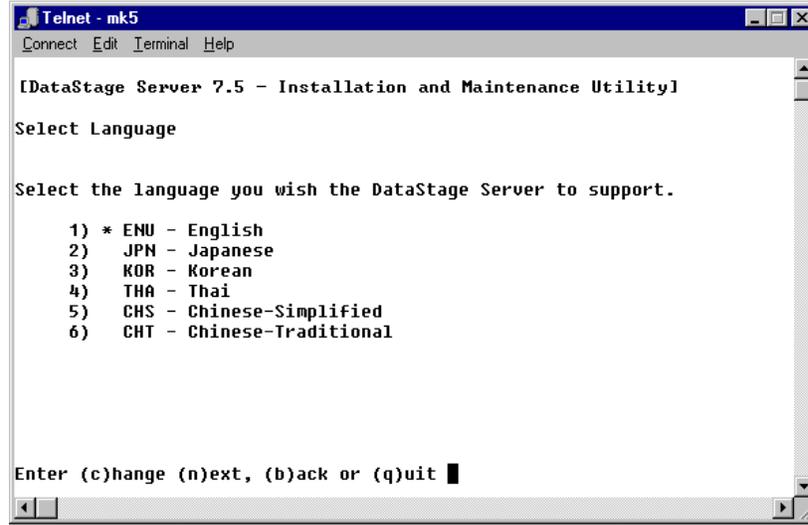
- 8 If you are upgrading from DataStage Release 5.1 or if you are upgrading from an earlier release and have chosen to upgrade your existing DataStage in step 4, you must install to the default location (which is where the existing DataStage is installed).

If you are upgrading from Release 4.x or earlier and have chosen to maintain your existing DataStage as a coresident UniVerse system in step 4, you must specify a different installation directory (<dsadmhome>/Ascential/DataStage is the normal location for 5.x installs).

To edit the location, select **(c)hange**, enter the new destination and press **Return**.

Select **(n)ext**. The NLS Select Language screen appears (NLS means National Language Support).

- 9 A list of supported languages appears, with the default language selected. Specify the server language to be supported.



To change the default selection, select **(c)hange**, select the appropriate language from the list of available languages, and press **Return**.

Select **(n)ext**. The National Language Support screen appears. This screen appears only if the server language selected is English. The default selection is set to **No**. (NLS is installed by default with all other languages)

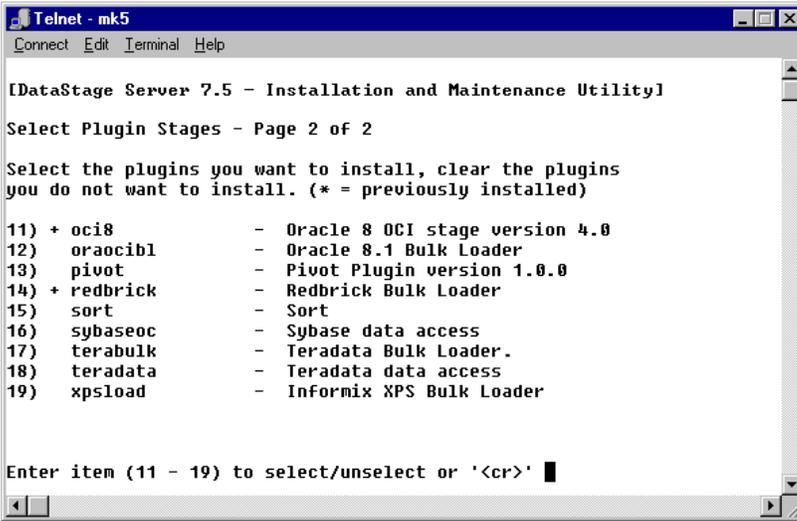
Select **(n)ext**. The Select Plug-in Stages screen appears.

- 10 Select **any additional** Plug-ins you want to install.

Do any of the following:

- Select **(a)ll** to select all Plug-ins.
- Select **(s)elect** to select individual Plug-ins.
- Select **(c)lear** to clear selected Plug-ins.
- Select **(p)age** to page through available Plug-ins.

If you choose select the screen changes to allow you to choose the Plug-ins you want to install.



```

Telnet - mk5
Connect Edit Terminal Help

[DataStage Server 7.5 - Installation and Maintenance Utility]
Select Plugin Stages - Page 2 of 2

Select the plugins you want to install, clear the plugins
you do not want to install. (* = previously installed)

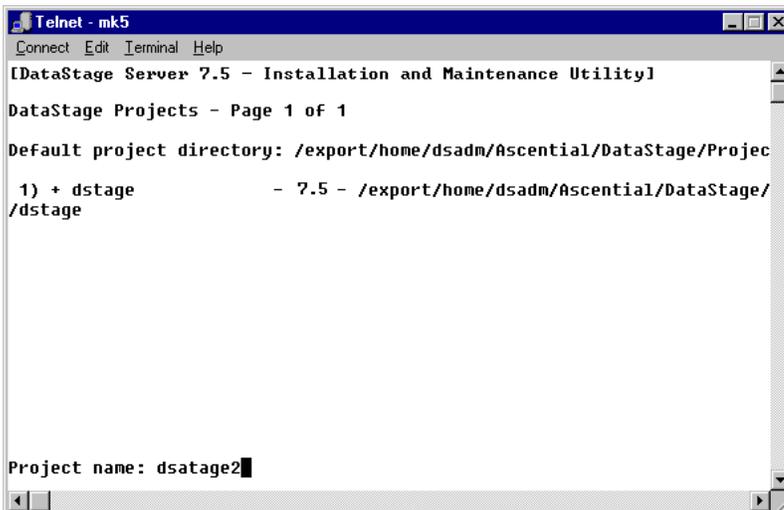
11) + oci8           - Oracle 8 OCI stage version 4.0
12) oraocib1        - Oracle 8.1 Bulk Loader
13) pivot           - Pivot Plugin version 1.0.0
14) + redbrick       - Redbrick Bulk Loader
15) sort            - Sort
16) sybaseoc        - Sybase data access
17) terabulk         - Teradata Bulk Loader.
18) teradata         - Teradata data access
19) xpsload          - Informix XPS Bulk Loader

Enter item (11 - 19) to select/unselect or '<cr>'
  
```

Type in the number of the Plug-in you want to select, a plus sign + appears against it in the list. Type its number again to deselect it. You can select as many Plug-ins as you require.

Existing Plug-ins will be upgraded automatically provided they were originally installed with the Install program (in install or maintenance mode). If they were installed with the Package Installer you will have to select them from the list.

Select **(n)ext**. The Projects screen appears.



```

Telnet - mk5
Connect Edit Terminal Help

[DataStage Server 7.5 - Installation and Maintenance Utility]
DataStage Projects - Page 1 of 1

Default project directory: /export/home/dsadm/Ascential/DataStage/Projec

1) + dstage          - 7.5 - /export/home/dsadm/Ascential/DataStage/
/dstage

Project name: dsatage2
  
```

- 11** The screen lists existing projects on your DataStage system. If you want to add more you can specify the names and locations of the projects you want to create. Do any of the following:

- Select **(a)dd** to add additional projects.
- Select **(c)hange** to change project definitions.
- Select **(d)elete** to delete project definitions.
- Select **(p)age** to page through defined projects.

Select **(n)ext**. If you licensed Enterprise Edition for the first time at this upgrade, the Choose Destination screen appears.

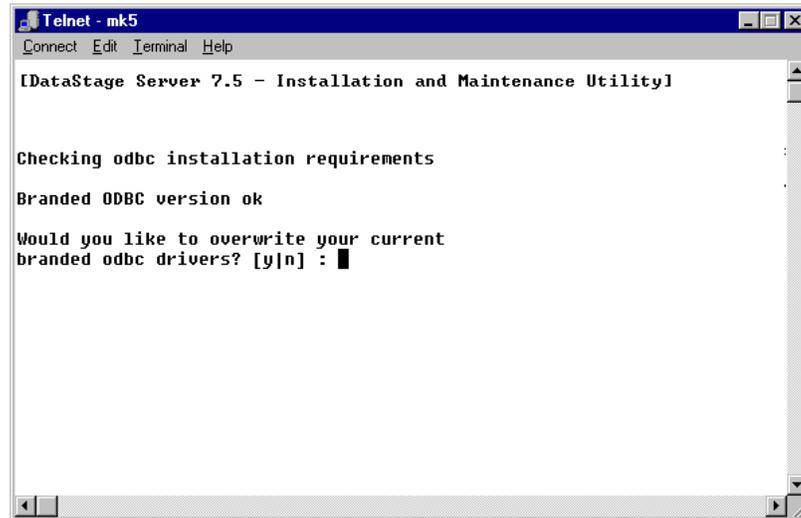
- 12** Select the location for the Parallel engine. Select **(n)ext**. The ORACLE HOME LOCATION - DataStage Enterprise Edition screen appears.
- 13** If you are intending to use Oracle, enter the Oracle home directory, otherwise leave it blank. Select **(n)ext**. The Specify SAS Version for use with DataStage Enterprise Edition screen appears.
- 14** If you are intending to use SAS, enter the version, otherwise leave it set to none. Select **(n)ext**. The DB2 Version screen appears.
- 15** If you are intending to use DB2, enter the version you want to access. Choose between 7, 8, or 'none' ('none' is the default). Select **(n)ext**. The Start Copying Files screen appears.
- 16** Select the platform extension packages you want to install. Do one of the following:
 - Select **(a)ll** to select all packages.
 - Select **(s)elect** to select individual packages.
 - Select **(c)lear** to clear selected packages.
 - Select **(p)age** to page through available packages.

Note This screen appears only if you have a valid Enterprise MVS Edition license, and optional platform extension packages are present on the CD-ROM.

Select **(n)ext**. The Start Copying Files screen appears.

- 17** The Start Copying Files screen displays information about the location, install type, NLS settings, project definitions, Plug-in packages, and optional platform extension packages if applicable. Do one of the following:
 - Select **(p)age** to page through and review the information.
 - Select **(b)ack** to go back and make changes.

Select **(n)ext**. The Checking odbc installation requirements screen appears.



- 18 If it is possible to preserve your existing ODBC drivers, then you are offered the choice of whether you want to overwrite your current drivers with the new Data Direct 4.2 drivers. Choose **y** to overwrite, or **n** to preserve.

Select **(n)ext** to start the upgrade process.

- 19 After installing Enterprise Edition you need to carry out the configuration tasks described in [Chapter 6, "Configuring for Enterprise Edition."](#)

Reinstalling and Maintenance Menu

To reinstall or access the maintenance menu, follow the instructions for an installation. If the installed version of DataStage matches the

CD-ROM version, the setup program reverts to the Maintenance mode screen as shown:

```

Telnet - mk5
Connect Edit Terminal Help

[DataStage Server 7.5 - Installation and Maintenance Utility]

Welcome to the DataStage Server Maintenance Utility.
This utility lets you modify the current installation

Please select the components you want to install (* = current selection)

1) Re-Install - repeat the last installation to restore
   missing files and settings.

2) * Add Projects - add new projects to the current installation.

3) Add Plugins - add new plugins to the current installation.

4) Copy Orchestrate - copy installation to other nodes

Enter (c)hange, (n)ext, or (q)uit █
  
```

This allows you to:

- Add new project definitions.
- Add new Plug-ins.
- Copy the parallel engine to other systems in MPP clusters.

- 1 Select **(c)hange** to enter change mode. Enter the corresponding number for the option you want (**1, 2, 3, or 4**). The asterisk (*****) indicates which option you are selecting. Press **Return**.
- 2 Select **(n)ext** and follow the on-screen instructions or select **(q)uit** to quit.

Copying Parallel Engine

Before you can copy the parallel engine to other platforms in a cluster system, you need to ensure that your machine has permission to run *rsh* on the target machines. Then proceed as follows, answering the questions from the copy script:

- 1 The Choose Source Location screen asks you to specify the path of the parallel engine on the current platform (cd to *\$DSHOME/./PXEngine* to find the full path name). Enter the path and select **(n)ext**. The install validates the path then goes onto the next step.
- 2 The Copy to nodes screen requests a list of nodes to which the parallel engine will be copied. Separate the node names with a space. Select **(n)ext** to go to the next step.
- 3 The Start Copying Files screen displays the copy details and asks for confirmation before it starts copying.

You can also perform this copy function from the UNIX command line if you prefer. See "[Copying the Parallel Engine to Your System Nodes](#)" on [page 6-3](#) for details.

Installation Log Files

Text output generated during the upgrade process is written to a disk based log file located in the `/tmp/dsinstall/logfiles` directory. Each log file contains information specific to a particular installation instance. Log files are generated for both clean and upgrade installations as well as maintenance mode functions. These log files can be viewed during the installation process using standard UNIX tools, such as `tail` or `more`.

Log file names are of the form `dsinstall.log.MMDDYY.HHMMSS` where `MMDDYY` equals the two digit month, day and year and `HHMMSS` equals the two digit hour, minute and seconds in 24 hour format, for example:

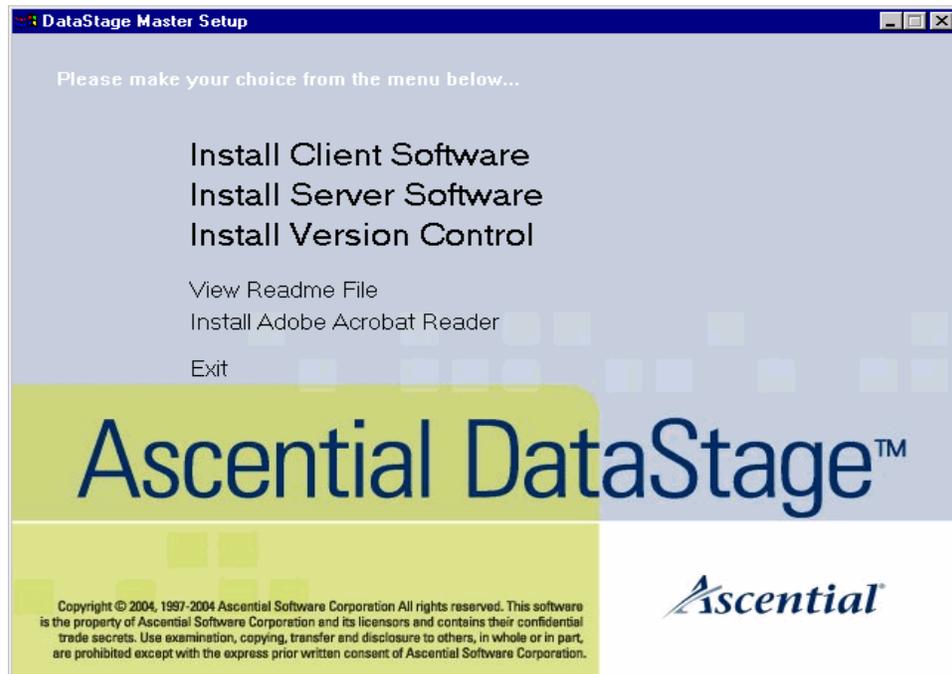
```
dsinstall.log.082201.162431
```

Upgrading DataStage Clients

DataStage clients are installed on Windows platforms (Windows 2000 or Windows XP). For each platform on which you want to upgrade the clients:

- 1 Insert the DataStage client CD-ROM. This contains a master setup program that manages the installation. The DataStage Master Setup screen should appear automatically. If Auto Run is switched

off and the screen does not appear, choose the CD-ROM drive from Windows **Explorer**, then double-click **install.exe**. The DataStage Master Setup screen is shown below.



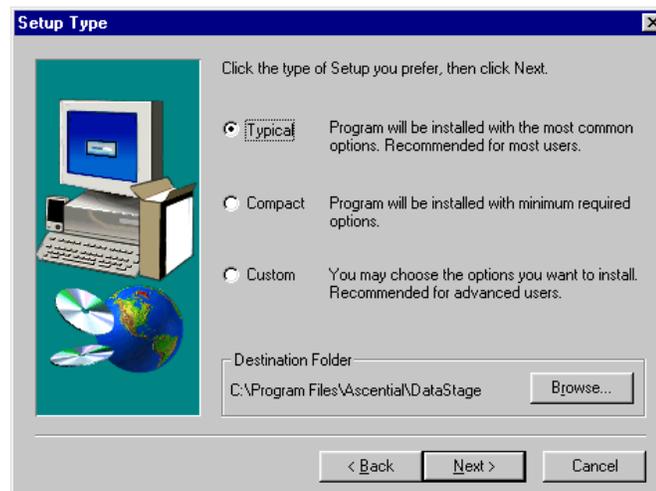
- 2 Choose the **Client** item. A setup program guides you through the install process:
 - a You are asked to enter your authentication details. On an upgrade the serial number, user count, and expiration date fields are filled in automatically, you just need to supply the authorization code.



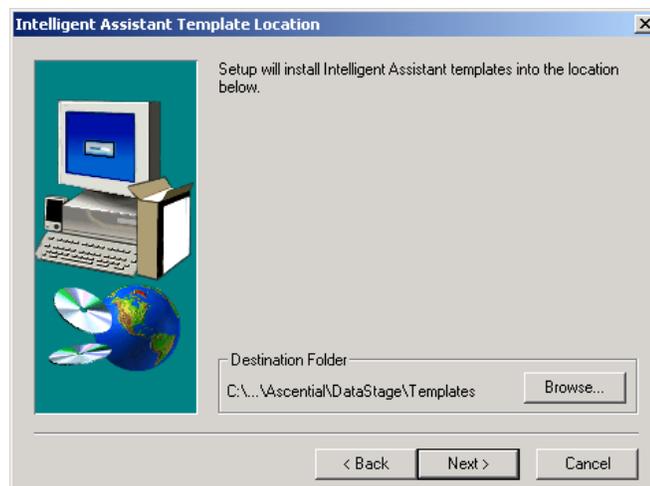
- b A dialog asks you if you want to upgrade the existing DataStage installation. If you choose **Yes**, your existing DataStage client will be overwritten by the new one. If you

intend to maintain multiple clients on your system and switch between them using the Multi-Client Manager, then you should choose **No**.

- c You are asked to choose between **Typical**, **Compact**, and **Custom** installs (**Typical** is recommended, but see the note below if you want to install the Multi-Client Manager). If you have chosen to overwrite the existing DataStage client, you should not change the **Destination Folder**, if you have chosen to preserve the existing client then you can accept the default destination for the new install (C:\Program Files\Ascential\DataStage7.5.1) or browse for a new destination



- d A screen asks you where you would like to install the Intelligent Assistant templates. Accept the default destination or browse for a new one.



The client install then proceeds. When it is finished it asks if you want to add shortcuts to your desktop, and whether you want to view the release notes.

3 For the latest information on clients, look at the release notes. In order to use the on-line documentation, you need to install the Adobe Acrobat reader.

Note If you want to install the Multi-Client Manager (which allows you to switch between different versions of DataStage client on the same machine), you should select the **Custom** installation mode and explicitly select the Multi-Client Manager.

Post-Upgrade Checks and Configuration

This section details steps you need to take post-upgrade to restore your existing DataStage configuration, or reconfigure if required. It includes the following steps:

- 1** Check your users and their permissions.
- 2** Configure for use with PAM, if required.
- 3** Check your backed up *dsenv* file and merge required contents with the newly installed *dsenv*, so that your environment variables are preserved.
- 4** Perform any configuration required for Plug-ins on your system.
- 5** If you have changed the *uvconfig* file to tune it for your system, you should merge in any changes into the new *uvconfig* file.
- 6** If you have chosen to preserve your existing ODBC drivers, you will need to restore the backed-up versions of *\$DSHOME/.odbc.ini*, *\$DSHOME/uvodbc.config*, and possibly the *uvodbc.config* file in each project directory. You will also need to merge data from your backed-up *dsenv* file with the new one.
- 7** If you have chosen to install the new ODBC drivers supplied with DataStage, you will need to configure them for use with your system.
- 8** If you have chosen to retain the UniVerse database from your previous DataStage installation as a coresident system you should verify that both the upgraded DataStage and the coresident UniVerse are operational.

Non-Root Upgrades

If you have performed a non-root upgrade, the following script must be run by the root user in order to enable DataStage:

```
$DSHOME/scripts/DSEservices.sh -add
```

This adds a service entry to */etc/services* for DataStage. (If you have installed as *root*, this is done automatically).

DataStage Users, UNIX Permissions and Security

If you are upgrading from versions earlier than release 7.0, DataStage projects are upgraded to have *dsadm* as owner and the permissions set to full access for all users which are a member of the same group as *dsadm*. When work is done in a project, additional files are created with the ownership and default permissions of the current user.

Permission problems result when default permissions for a user prevent other users from accessing important files and directories in the project. To best solve permissions problems and maintain tight UNIX security, ensure that DataStage users have the same primary group as *dsadm*. All DataStage users must be members of this group. We recommend calling this group *dstage*.

On System V platforms (AIX, HP-UX, and Solaris), you may also set SGID access on the project directories. Thus, files and directories created in the project directory belong to the same group as the project directory. See your UNIX documentation for more information on SGID (see manual page for CHMOD).

Configuring for Use with PAM

From DataStage 7.0 onwards you can, if required, use an authentication mechanism other than the standard UNIX one. To this end, DataStage supports PAM (pluggable authentication modules). This provides a way of keeping underlying authentication technologies separate from application code, thus eliminating the need to update the application every time the authentication mechanism is changed.

PAM is currently supported on Solaris, HP, AIX, and LINUX platforms. The implementation on each platform is slightly different, and you should consult the documentation supplied with your platform for details. It is assumed you have a working knowledge of PAM and the various authentication modules and strategies.

To implement PAM support on DataStage, you need to:

- Set the AUTHENTICATION flag in the DataStage system file *uvconfig*.
- Add the PAM service entry, *dsepam*, to the PAM configuration file.

DataStage ships with PAM support disabled, and the AUTHENTICATION flag set to 0.

The name and location of the PAM configuration file is platform dependent. For example, versions of PAM of the type generally found on a Solaris system use a PAM configuration file *pam.conf* located in the */etc* directory, and you add entries to this. Versions of Pam of the type generally found on a LINUX system have a specific file for each application located in the */etc/pam.d* directory.

The specific steps you need to add PAM support are:

- 1 Add to or create the PAM configuration file on your platform.
- 2 Stop the DataStage server engine as follows:
- 3 Edit *\$DSHOME/uvconfig* and set the AUTHENTICATION flag to 1:

```
$DSHOME/bin/uv -admin -stop
```

```
# AUTHENTICATION - Specifies the method by which UNIX user
# authentication is done. Currently, the following methods
# are supported:
#
#    0) Standard O/S Authentication (default)
#    1) Pluggable Authentication Module (PAM)
#
# This value should only be changed with a full understanding
# of the implications, as improper setting of this value can
# lead to the environment being unusable.
```

```
AUTHENTICATION 1
```

- 4 Regenerate the DataStage configuration as follows:
- 5 Restart the DataStage server engine as follows:

```
$DSHOME/bin/uv -admin -regen
```

```
$DSHOME/bin/uv -admin -start
```

Example PAM Configuration Files

On a LINUX system you would need to create a file called *dsepam* in the */etc/pam.d* directory. This would contain entries similar to the following:

```
##%PAM-1.0
auth    required /lib/security/pam_stack.so service=system-auth
passwordrequired /lib/security/pam_stack.so service=system-auth
account required /lib/security/pam_stack.so service=system-auth
```

On a Solaris system you would need to edit the existing *pam.cof* file in the */etc* directory and add an entry similar to the following:

```
dsepam  auth required  /usr/lib/security/pam_unix.so.1
```

Environment Variables

DataStage has a centralized file for storing environment variables called *dsenv*. It resides in *\$DSHOME*, where *\$DSHOME* is the

DataStage main directory (for example `/u1/dsadm/Ascential/DataStage/DSEngine`). (Versions of DataStage prior to release 7.0 used a file called `/.dshome` to identify the DataStage main directory.)

To restore the environment variables of your existing DataStage installation, you need to merge the old `dsenv` that you backed up with the new `dsenv` installed with the upgrade. You can use the backed up file as the basis for the new `dsenv` but you should copy the following lines from the new `dsenv` file and overwrite the equivalent lines in your old `dsenv`:

```
LD_LIBRARY_PATH=$DSHOME/lib:/.dslibs:$LD_LIBRARY_PATH; export
LD_LIBRARY_PATH
```

This environment variable specifies the shared library path and its actual name is platform dependent as follows:

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

If you have added new Plug-ins during your install, or if you are intending to use ODBC drivers to connect to new databases you will probably have to add some new environment variables to your `dsenv` file.

To emulate the DataStage server environment, in a Bourne shell execute the following command from the `<dshome>/DSEngine` directory:

```
. ./dsenv
```

Once you have done this, you can refer to the `<dshome>` directory as `$DSHOME`

After changing or adding any environment variables, the DataStage server should be stopped and restarted as follows:

- 1 To stop the server:
`$DSHOME/bin/uv -admin -stop`
- 2 To start the server:
`$DSHOME/bin/uv -admin -start`

Ensure that you allow sufficient time between executing stop and start commands (minimum of 30 seconds recommended).

Configuring Plug-ins

DataStage provides a number of Plug-ins which are used to access particular databases and perform other special functions. Any Plug-ins you had previously installed are automatically upgraded. If you have installed additional Plug-ins, you may need to specify some environment variables in *dsenv*. Up to date information about environment variables are in the technical bulletins supplied with each Plug-in and in *DataStage Plug-in Installation and Configuration Guide*. We advise that you check the documentation for upgraded Plug-ins too, in case requirements have changed. These are accessible from the main PDF contents list (open this by choosing **Ascential DataStage ► Online Manuals ► DataStage Documentation** from the Start Menu). They are in Acrobat format and require an Acrobat reader.

Note Some of the Plug-in technical bulletins still contain references to the special file called */.uvhome*. This file determines where DataStage is installed. At release 5.2 onwards the DataStage installation directory is identified by a new file called */.dshome*. So for */.uvhome* read */.dshome*.

BCPLoad Plug-In

The following details apply to the BCPLoad Plug-in, which is installed with DataStage by default. The BCPLoad Plug-in enables you to use the bulk copy program (BCP) on SQL Server and Sybase databases.

The BCPLoad stage uses the BCP API in client libraries. The BCPLoad Plug-in does **not** support the following libraries:

- MSDBLIB as a client library on any UNIX platform.
- SYBCTLIB as a client library on an HP-UX platform.

Note that Sybase DBLIB is no longer supported on Solaris or AIX platforms, use CTLIB instead.

When using a BCPLoad stage with a UNIX server, you must specify SYBCTLIB or SYSDLIB as client libraries. This is done by setting the **Client Library** property on the Stage editor **Properties** page. When using a BCPLoad stage with an HP-UX server, you must specify SYSDLIB as the client library.

Configuring ODBC Access

DataStage also allows you to access any database that supports ODBC using database-specific ODBC drivers. DataStage comes complete with a set of ODBC drivers for this purpose which are installed automatically. During the upgrade you were offered a choice of upgrading drivers or maintaining your existing set. The action you need to take depends on which option you chose:

If you chose to retain your old drivers you need to:

- 1 Take back ups of *\$DSHOME/dsenv* and *\$DSHOME/odbc.ini* before you upgrade.
- 2 Identify the path where your existing odbc drivers are installed (usually *\$DSHOME/./branded_odbc*).
- 3 When upgrading, choose to preserve your old ODBC drivers when prompted. The existing *branded_odbc* directory is renamed *branded_odbc.xxx* during upgrade, where *xxx* is the previous odbc release number.
- 4 Preserve the newly installed *dsenv* and *.odbc.ini* files by renaming them with a suitable suffix (e.g. *.ship*).
- 5 Reinstate the existing ODBC drivers by renaming the saved directory *branded_odbc.xxx* back to the path you noted in step 2).
- 6 Manually merge the backed-up *dsenv* file with that saved at step 4).
- 7 Reinstate the original *odbc.ini* file.
- 8 Ensure that all paths in the *.odbc.ini* file are still correct.
- 9 Ensure that the *.dslibs* link points to the lib directory within the path noted at step 2) above.

If you have retained the old drivers and subsequently wish to configure a new ODBC connection using these drivers, you need to follow the configuration guide for non-wire protocol drivers in the following sections.

Upgraded ODBC Drivers

Note ODBC driver packs are often updated between major releases of DataStage. You are strongly advised to check your release notes for more up-to-date information about ODBC drivers.

Ascential provides an OEM version of the Data Direct ODBC driver pack with this release. These drivers are licensed solely for use with DataStage and require certain connection parameters to be set in

order to function properly. Do not use these drivers with other applications, as licensing errors result.

The ODBC drivers are one of two types, depending on the database being connected to and your platform type:

- non-wire protocol drivers - these require you to install the client software for the database on the DataStage server (the drivers use the API supplied by the database client)
- wire protocol drivers - these do not require database client software (they communicate with the database directly)

The following table lists the the drivers installed with DataStage, which databases they access, and which platforms they are available on. Please see the release notes for information about which ODBC drivers and database versions have been validated at this release of DataStage.

Driver	Database	Platform
DB2 Wire Protocol (VMdb220.so)	DB2 wire protocol UDB 7.1, UDB 7.2, UDB 8.1 DB2 for OS/390 6.1, 7 DB2 for OS/390 and z/OS 7 DB2 for AS/400 V4R5, V5R1, V5R2	Solaris, AIX, LINUX, HP-UX
dBase (VMdbf20.so)	dbase IV, V FoxPro 3.0, 6.0 (3.0 functionality level)	Solaris, AIX, LINUX, HP-UX
Informix Client Driver (VMifc120.so)	Informix OnLine 7.x Informix SE 7.2x Informix Dynamic Server with Universal data Option 9.1.4+ Informix Dynamic Server 9.2x Informix Dynamic Server 9.3x	Solaris, AIX, LINUX, HP-UX
AIX 5.1 Informix Client Driver (VMinf5120.so)	Informix OnLine 7.x Informix SE 7.2x Informix Dynamic Server with Universal data Option 9.1.4+ Informix Dynamic Server 9.2x (this driver requires Informix 2.70UC3 SDK or higher for AIX 5.1 to be installed on the DataStage Server)	AIX 5.1
Informix Wire Protocol (VMinf20.so)	Informix Dynamic Server with Universal Data Option 9.1.4+ Informix Wire Protocol Informix Dynamic Server 9.2x Informix Dynamic Server 9.3x	Solaris, AIX, LINUX, HP-UX

Informix Wire Protocol (VMinf19.so)	Informix Dynamic Server with Universal Data Option 9.1.4+ Informix Wire Protocol Informix Dynamic Server 9.2x Informix Dynamic Server 9.3x	Tru64
Oracle (VMor820.so)	Oracle 7.3.4+, 8.0.5+ Oracle 8i R1, R2, R3 (8.1.5, 8.1.6, 8.1.7) Oracle 9i R1, R2 (9.0.1, 9.2)	Solaris, AIX, LINUX, HP-UX
Oracle Wire Protocol (VM0ra20.so)	Oracle 8i R2, R3 (8.1.5, 8.1.6, 8.1.7) Oracle 9i R1, R2 (9.0.1, 9.2)	Solaris, AIX, LINUX, HP-UX
Oracle Wire Protocol (VM0ra19.so)	Oracle 8i R2, R3 (8.1.5, 8.1.6, 8.1.7) Oracle 9i R1, R2 (9.0.1, 9.2)	Tru64
Progress (VMpro120.so)	PROGRESS 9.1B, 9.1C via SQL92 engine	Solaris, AIX, LINUX, HP-UX
SQLServer Wire Protocol (VMmsss20.so)	MS SQLServer 7.0 MS SQLServer 2000 (with service packs 1, 2 and 3)	Solaris, AIX, LINUX, HP-UX
SQLServer Wire Protocol (VMmsss19.so)	MS SQLServer 7.0 MS SQLServer 2000 (with service packs 1, 2 and 3)	Tru64
Sybase Wire Protocol (VMase20.so)	Sybase Adaptive Server 11.5+ Sybase Adaptive Server Enterprise 12.0 and 12.5	Solaris, AIX, LINUX, HP-UX
Sybase Wire Protocol (VMase19.so)	Sybase Adaptive Server 11.5+ Sybase Adaptive Server Enterprise 12.0 and 12.5	Tru64
Text (VMtxt20.so)	Text files	Solaris, AIX, LINUX, HP-UX

Configuring ODBC Connections

You need to edit three files to set up the required ODBC connections. These are:

- dsenv
- .odbc.ini
- uvodbc.config

All three are located in the *\$DSHOME* directory. (Copies of *uvodbc.config* are also placed in the project directories – see "[uvodbc.config File](#)" on [page 5-28](#).)

Non-wire drivers require different set-up information to wire drivers. Non-wire drivers require information about the location of the database client software, wire drivers require information about the database itself.

For more information on configuring the ODBC environment for your specific DBMS, see the Data Direct Drivers Reference manual *odbceref.pdf* file located in the *\$DSHOME/./branded_odbc/books* directory. You should also check the ODBCREAD.ME file in the *branded_odbc* directory. There is also an html file located in *branded_odbc/odbchelp*.

Once you have configured the required drivers, you should test connectivity as described in "[Testing ODBC Driver Connectivity](#)" on page 5-30.

***dsenv* File**

Any environment variables you need for interactive use of ODBC drivers to make a connection to an ODBC data source must be added to the *dsenv* file. This lets the DataStage server inherit the proper environment for ODBC connections.

For a connection using a wire protocol driver, there are no changes required to *dsenv*.

For a connection using a non-wire protocol driver, you generally need to specify the following in the *dsenv* file:

- environment variables required by the database client software
- database home location
- database library directory

The following gives typical entries for commonly used databases:

```
#Sybase 11 setup
LANG=
Export LANG
SYBASE=/export/home/sybase/sybase;export SYBASE
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$SYBASE/lib:/usr/lib:/lib;export
LD_LIBRARY_PATH
# Oracle 8i
ORACLE_HOME=/space/oracle8i
ORAHOME=/space/oracle8i
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib:$ORACLE_HOME/rdbms/
lib;export LD_LIBRARY_PATH
ORACLE_SID=WSMK5
ORASID=WSMK5
export ORACLE_HOME ORAHOME ORACLE_SID ORASID
#informix XPS 9.3
INFORMIXDIR=/space/informix; export INFORMIXDIR
INFORMIXSERVER=so1new2.1;export INFORMIXSERVER
ONCONFIG=onconfig.so1new2; export ONCONFIG
```

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$INFORMIXDIR/lib:$INFORMIXDIR/lib/
esql:$INFORMIXDIR/lib/cli;export LD_LIBRARY_PATH
LANG=C
export LANG
#DB2 6.1
DB2DIR=/opt/IBMDB2/V6.1;export DB2DIR
DB2INSTANCE=DB2inst1; export DB2INSTANCE
INSTHOME=/export/home/DB2inst1;export INSTHOME
PATH=$PATH:$INSTHOME/sql1lib/bin:$INSTHOME/sql1lib/adm:$INSTHOME/sql1lib/
misc
export PATH
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$INSTHOME/sql1lib/lib;export
LD_LIBRARY_PATH
THREADS_FLAG=native;export THREADS_FLAG
```

You should stop and restart the DataStage server whenever you edit *dsenv*:

1 To stop the server:

```
$DSHOME/bin/uv -admin -stop
```

2 To start the server:

```
$DSHOME/bin/uv -admin -start
```

Ensure that you allow sufficient time between executing stop and start commands (minimum of 30 seconds recommended).

***.odbc.ini* file**

The *.odbc.ini* files gives information about connecting to the Database (wire protocol drivers) or the database client (non-wire protocol drivers). If your system uses a mix of drivers, your *.odbc.ini* file will contain a mix of entry types.

Configuration examples for the various platforms are provided in *branded_odbc/Ascential_Tools* directory as follows:

- *aix_4.2.odbc.ini* for AIX systems.
- *sun_4.2.odbc.ini* for SunOS systems.
- *linux_4.2.odbc.ini* for LINUX systems.
- *hpux_4.2.odbc.ini* for HPUX systems.

These files give the minimum parameter settings required to create a working *odbc.ini* file. They each give configurations for the following databases:

- DB2 (UNIX, Windows, AS400)
- Oracle (UNIX)
- Informix (UNIX and Windows)
- MS SQL Server 200 connection

■ Sybase (Windows)

***uvodbc.config* File**

This file specifies the DSNs for the databases you are connecting to via ODBC. Entries in *uvodbc.config* have the form:

<name>

```
DBMSTYPE = ODBC
```

Name is the ODBC data source name (DSN), this should be the same name as specified in the *.odbc.ini* file. Note that the spaces either side of the equal sign are required.

A typical *uvodbc.config* file is as follows:

```
[ODBC DATA SOURCES]

<localuv>
DBMSTYPE = UNIVERSE
network = TCP/IP
service = uvserver
host = 127.0.0.1
<Sybase1>
DBMSTYPE = ODBC
<Sybase2>
DBMSTYPE = ODBC
<Oracle8>
DBMSTYPE = ODBC
<Informix>
DBMSTYPE = ODBC
<DB2>
DBMSTYPE = ODBC
```

A copy of the *uvodbc.config* file is also placed in each DataStage project directory (for example, */ui/dsadm/Ascential/DataStage/Projects/Dstage1*). 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 the file in *\$DSHOME*. If you alter *uvodbc.config* after creating projects you should copy the edited file to the project directories.

DB2 Connections

Before connecting to a DB2 DSN you need to bind DB2 packages to the DSN. The method differs according to whether you are using a wire protocol driver, or a non-wire protocol driver:

Wire Protocol Driver.

You need to bind a package to every DSN you are going to use to connect to a DB2 database. To create and bind a package to a DSN, from the `$DSHOME/./branded_odbc/lib` directory enter:

```
bind20 <dsn>
```

Where `<dsn>` is the ODBC data source name. You are prompted for a username and password if one is required.

Third Party ODBC Drivers

You can also use ODBC drivers from third parties for connecting DataStage to different data sources. You might require to connect to a database not supported by the Data Direct driver pack, for example, Redbrick. You should be aware that you can only use one ODBC Driver Manager at a time, however. This means you cannot use third party drivers at the same time as you use the DataStage drivers, and you have to perform some re-configuration to switch between the two types.

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.

Note To use an ODBC driver on an AIX platform other than one of the supplied Data Direct ones, contact Ascential support.

To use a third party ODBC driver:

- 1 Install the ODBC manager and driver(s) on your UNIX platform following the instructions supplied by the driver vendor.
- 2 Configure DataStage to connect to the driver. This involves:
 - a Adding required environment variables to the `dsenv` file.
 - b Setting up the connection to database client or database in the `.odbc.ini` file (must specify DSN, driver name, and log in information as a minimum).
 - c Specify the DSN for connecting to the database in the `uvodbc.config` file.

General information about editing `dsenv`, `.odbc.ini` and `uvodbc.config` are given in "[Configuring ODBC Connections](#)" on [page 5-25](#). Details of the exact information to add should be supplied in the driver vendor literature.

- 3 Include the shared library `libodbc.xx` (the suffix `xx` is either `sl` or `so` depending on platform) for the third party ODBC Manager in the DataStage load library path. Do this by editing `dsenv` and ensuring that the path to the third party Manager precedes that to

`$DSHOME/./branded_odbc/lib` in the shared library path environment variable (see ["Environment Variables"](#) on [page 3-28](#) for details of `dsenv` and the shared library path environment variable).

- 4 Once you have installed and configured your driver, you should test connectivity as described in ["Testing ODBC Driver Connectivity"](#) on [page 5-30](#).

To switch back to using Data Direct drivers you must reverse the procedure in step 3, that is ensure that the DataStage ODBC driver library precedes any third party driver libraries.

Testing ODBC Driver Connectivity

You can test the connectivity of the ODBC connections you have defined for DataStage as follows:

- 1 If you haven't previously done so, `cd` to `$DSHOME` and set up the DataStage environment by running `dsenv`:

```
./dsenv
```

- 2 Invoke the DataStage engine shell:

```
./bin/dssh
```

The DSEngine shell starts.

- 3 Log to the project:

```
LOGTO project_name
```

Where *project_name* is case sensitive.

- 4 Get a list of available DSNs by typing:

```
DS_CONNECT
```

- 5 Test the required connection by typing:

```
DS_CONNECT DSN
```

Where *DSN* specifies the connection you want to test. Enter username and password as requested, you should then be connected to the specified database.

- 6 Enter `.Q` to quit the connection.

If the connection does not work, consult ["ODBC Connection Problems"](#) on [page 10-3](#) for guidance.

Upgrading Non-Root Installations

If you upgraded DataStage using an id other than `root` (see ["Types of Upgrade"](#) on [page 5-30](#)), you can subsequently upgrade DataStage to

have one or more of the features of a *root*-installed system. You do this by running a set of scripts. These scripts must be run by *root*, but your system administrator can examine the scripts and assure themselves that they do not compromise security in any way.

The features that you might want to enable in this way are:

- Auto startup (DataStage services restart automatically when the machine restarts).
- Impersonation mode (DataStage server will run with the id and permissions of the user who logged in).

The scripts used to enable or disable these features are:

- *DSEnable_autostartup.sh*. Enables dsengine auto startup at system boot time.
- *DSEdisable_autostartup.sh*. Disables dsengine auto startup at system boot time.
- *DSEnable_impersonation.sh*. Enables impersonation features.
- *DSEdisable_impersonation.sh*. Disables impersonation features.
- *DSEsetperms.sh*. Called by *DSEnable_impersonation.sh* to set permissions.
- *DSEresetperms.sh*. Called by *DSEdisable_impersonation.sh* to reset permissions.
- *DSEservices.sh*. Used to add or update dsrpc entry in */etc/services* file. This must be run on non-root installed systems to enable DataStage (see "[Post-Install Checks and Configuration](#)" on [page 3-26](#)).

You can establish the current operational state of the system by issuing the command:

```
$DSHOME/bin/uv -admin -info
```

The scripts are located in *\$DSHOME/scripts*.

You can also enable or disable the auto start and impersonation features using the `uv -admin` command as follows:

- `uv - admin -autostart on`. Enable auto start.
- `uv - admin -autostart off`. Disable auto start.
- `uv - admin -impersonation on`. Enable impersonation.
- `uv - admin -impersonation off`. Disable impersonation.

6

Configuring for Enterprise Edition

This chapter tells you how to set up your user environment on a UNIX platform to run parallel jobs.

"The Parallel Engine Configuration File" in *Parallel Job Developer's Guide* describes the configuration files that specify what processing, storage, and sorting facilities on your system should be used to run a parallel job.

Your System's Remote Shell

The parallel engine must be able to run the remote shell command *rsh* without a password on all processing nodes running parallel jobs.

Specifying the Location of *rsh*

To find *rsh* on a processing node, the parallel engine searches the following paths in the order given:

- install_dir/etc/remsh* (if it exists)
- /user/lpp/ssp/rcmd/bin/rsh* (AIX only)
- /usr/ucb/rsh*
- /usr/bin/remsh*
- /bin/remsh*
- /usr/bin/rsh*

where *install_dir* is the top-level directory of your parallel engine installation.

If the parallel engine does not find your *rsh* command, you must specify its location. To do so, copy or rename the supplied file

install_dir/etc/remsh.example to *install_dir/etc/remsh*. This file contains the following shell script:

```
#!/bin/sh
# Example apt/etc/remsh
exec /usr/bin/rsh "$@"
```

As written, this shell script invokes */usr/bin/rsh*. Edit the last line of this script to invoke your specific remote shell command. The script should be executable by all users. Use *chmod* to ensure that it is:

```
# chmod 755 script-filename
```

Test this by running *rsh* on each node:

```
# rsh nodename uptime
```

Allowing User Execution of *rsh* Without a Password

You perform this process differently according to the type of system you are running. For example, you can either edit */etc/hosts.equiv* or create a *.rhosts* file for each user. In both cases, add the host name of each Parallel processing node to */etc/hosts.equiv* or *.rhosts*, one host name per line.

The host name that is included in this file must correspond to the setting of the node's *fastname* parameter in the Parallel configuration file. For information on the *fastname* configuration option, see "[Node Names](#)" in *Parallel Job Developer's Guide*.

If you choose to edit */etc/hosts.equiv*, the file must be owned by *root* and must grant read/write access to *root* and no access to any other user (file mode of 600).

If you choose to create an *.rhosts* file for each user, it must:

- Be located in the home directory of each Parallel user.
- Be owned by the user.
- Grant read/write access to the user and no access to any other user (file mode of 600).

To check that users can use *rsh* without a password, issue the following command on each node:

```
$ rsh hostname uptime
```

where *hostname* is the name of a processing node that you use with the parallel engine. If *hostname* is accessible, this command prints a message displaying the time it has been up.

Copying the Parallel Engine to Your System Nodes

Note This step applies to MPP and network-cluster systems only.

You can use the DataStage maintenance menu to copy the parallel engine to system nodes (see [page 3-22](#) or [page 5-13](#)). Alternatively, you can use the script described here to copy the parallel engine to other nodes.

To copy the parallel engine to all processing nodes:

1 Make the parallel engine accessible to your chosen systems using one of the following methods:

- Globally cross-mount, typically via NFS, a single directory on a single system containing the parallel engine software. This configuration makes software upgrades more convenient than if the parallel engine is installed on all processing systems. If you are using NFS to globally mount the directory, mount it using the *hard* or the *hard, intr* option. Do not mount it using the *soft* option.

or

- Copy the parallel engine to a directory with the same path name on all processing systems that you designate for processing parallel jobs. You may notice faster startup times with this configuration than if you globally cross-mount.

2 Invoke the parallel engine *copy-orchdist* utility to copy the installation to all your new nodes. The necessary directories will be created on each node:

```
node: $ install_dir/install/copy-orchdist node1 node2 ... nodeN
```

where *install_dir* is the top-level directory of your parallel engine directory, and *node1, node2 ... nodeN* is a space-separated list of the new node names.

If you later add nodes to your system, you must run the *copy-orchdist* utility again to copy the parallel engine to the new nodes, as in this example:

```
node: $ install_dir/install/copy-orchdist newNode1 newNode2 ...  
      newNodeN
```

In addition, you must either create a new configuration file defining the new nodes or add them to an existing configuration file. See "[The Parallel Engine Configuration File](#)" in *Parallel Job Developer's Guide*.

Setting the wall System Parameter

Note This procedure is necessary only for AIX MPP and network clusters.

If you are installing the parallel engine on an RS/6000 SP or a network of workstations, you must verify the setting of the network parameter *thewall*. The value of this parameter can greatly affect the performance of the parallel engine.

Set *thewall* to at least 25% of each node's physical memory, or the maximum allowed on your system, if that is less than 25% of memory. The maximum value of *thewall* is OS-version dependent. The man page for the network options (*no*) command contains the details and system default values.

The value of *thewall* is specified in kilobytes. For example, if each node on your system has 256 MB (262,144 KB) of physical memory, set *thewall* to 65,536.

To set *thewall*:

- 1 Determine the amount of physical memory on a node (the value of *realmem* is the amount of physical memory on the node in KB):

```
node: $ lsattr -E -l sys0 | grep realmem
```

- 2 Determine the current setting of *thewall* for a particular node or workstation:

```
node: $ /usr/sbin/no -a | grep thewall
```

- 3 Set *thewall* by doing one of the following:

- On a specific node, execute the following command with *root* privileges:

```
node: # /usr/sbin/no -o thewall=65536
```

or

- Set *thewall* on all nodes of an RS/6000 SP system by executing the following *dsh* command from the control workstation, with *root* privileges:

```
control: # dsh -a no -o thewall=65536
```

Setting the Maximum Number of Processes

A Parallel user should be able to run at least 100 processes per node and possibly more on large systems or SMPs.

To configure the number of processes for parallel jobs:

- 1 Determine the current setting for the number of processes, as defined by the setting of *maxuproc*, by issuing the following command:

```
node: $ lsattr -E -l sys0 | grep maxuproc
```

- 2 If *maxuproc* is less than 100, do one of the following:

- On a specific node or workstation, issue the following command to set the maximum number of processes. This command must be executed on the node with root access privilege:

```
node: # chdev -l sys0 -a maxuproc=100
```

or

- On all nodes of an AIX RS/6000 SP system, issue the following *dsh* command to set the maximum number of processes per user. This command must be executed from the control workstation with root privilege:

```
control: # dsh -a chdev -l sys0 -a maxuproc=100
```

or

- On AIX, you can also issue the *smit* command to perform this action.

Setting the Maximum Amount of Memory on AIX

Because parallel engine processes consume large amounts of memory, you should ensure that you have permission to use sufficient resources to run your parallel jobs. There are a number of parameters that you can set to control memory resources. The most important parameters are:

- **fsize.** Specifies the largest file a user may create. Specify 2 GB minus 512 bytes (4,194,303 512-byte blocks).
- **data.** Specifies largest data segment (sometimes called heap) size a program may have. Specify 128 MB (262,144 512-byte blocks) or more, or -1 to specify no limit.
- **stack.** Specifies the largest stack size a program may have. Specify 32 MB (65,536 512-byte blocks) or more.
- **rss.** Specifies the maximum amount of physical memory a user's process may use. Specify at least 64 MB (131,072 512-byte blocks) or more (unlimited).

To set the memory resource parameters for each user:

- 1 Determine the value of the memory size parameters for a user with the command:

```
lsuser -f user
```

This command displays various parameters, including:

```
fsize=2097151
data=262144
stack=65536
rss=131072
```

These numbers denote quantities of 512-byte blocks.

- 2 Set the parameters for the user as defined by the requirements listed above.

To set *fsize* to 4,194,303 512-byte blocks (2 GB minus 512 bytes), issue the following command on a single node or workstation using *root* access privileges:

```
node: # chuser fsize=4194303 user1
```

Issue the following command to set the parameter on all nodes of an AIX RS/6000 SP. This command must be issued from the control workstation using root access privileges.

```
control: # dsh -a chuser fsize=4194303 user1
```

After this command is completed, users must log out and in again for it to take effect.

To assign values to the other parameters, use the syntax shown above, replacing the parameter name and value with the new parameter name and value.

Setting the Per-User Resource Limits on Compaq Tru64

Set per-user resource limits to control virtual address space, maximum user processes, and per-process data size.

The command:

```
$ /sbin/sysconfig -q proc max_proc_per_user
```

controls how many processes an individual user can run. Use a value of 1000 for larger parallel job flows.

These two commands:

```
$ /sbin/sysconfig -q proc per_proc_data_size
```

```
$ /sbin/sysconfig -q proc max_per_proc_data_size
```

control the per-process allocatable memory. They should be set to 1 G or more depending on the memory patterns of your operators.

Use *sysconfigdb(8)* to edit the persistent parameters table (*/etc/sysconfigtab*). One way to do this is to create an input file, use it to add the items to the database, then re-synchronize the running kernel to it to make the changes active. To do this, create a new file called, for example, *sysconfig.new*, containing these lines:

```
proc:
    max_proc_per_user = 1000
    max_per_proc_address_space = 1073741824
    per_proc_address_space = 6442450944
```

Then run the following commands as *root*:

```
# cd /etc
# /sbin/sysconfigdb -a -f sysconfig.new
# /sbin/sysconfig -r proc
# /sbin/sysconfig -r vm
```

After making these changes, log out and log in again. The results of running the command:

```
$ ulimit -a
```

should reflect the new settings.

Increasing the Limits on Segment Sizes on HP-UX

Specify the maximum size of segments, including data and text segments, to the values shown here:

```
maxdsiz = 2039480320 (1945 MB)
maxssiz = 82837504 (79 MB)
maxtsiz = 1073741824 (1024 MB)
```

You can examine the current values of these parameters using the command:

```
$ /usr/sbin/sysdef | grep parameter-name
```

Note that the values returned by *sysdef* are numbers of 4-KB pages, not the values in bytes.

These parameters can be set via *sam*, the System Administration Manager, or by modifying the file */stand/build/system*. After changing these values you must rebuild the kernel. Use the command *Process New Kernel* in *sam* and reboot the system.

Setting the System Paging Space

For parallel jobs that process large amounts of data, your system should have at least the following settings for the paging space:

- 500 MB per processor for systems with fewer than 20 processors
- 1 GB per processor for systems with 20 or more processors

Depending on your parallel job designs you may find your requirements are greater. The more stages an individual job has, the greater the swap space requirement is likely to be. Similarly, you are likely to need more swap space if you intend to run more than one job at once.

Note Only a system administrator with *root* privileges should reset the system swap space. The procedure is potentially destructive.

Setting the Paging Space on AIX

- 1 Determine the current paging size for a node or workstation by issuing this command:

```
node: $ lsps -a
```

You must run the *lsps* command on each node or workstation in your system.

The *lsps* command displays information in the form shown in this table.

Page Space	Physical Volume	Volume	Group Size	% Used	Active	Auto	Type
paging01	hdisk1	pagevg	2000MB	1	yes	yes	lv
paging00	hdisk0	rootvg	192MB	5	yes	yes	lv
hd6	hdisk0	rootvg	64MB	23	yes	yes	lv

The Group Size column lists the paging size for each disk drive. In this case, the total for the node is 2000 + 192 + 64 MB = 2256 MB (approximately 2.2 GB).

You can also issue the following *dsh* command from the control workstation of an AIX RS/6000 SP to display paging size information about all nodes: *control: \$ dsh -a lsps -a*

- 2 Change the paging space as necessary for each node or workstation. See the *IBM System Management Guide: Operating System and Devices* manual for more information.

Setting the Paging Space on Sun Solaris

Although any user can determine how much swap space there is, only *root* can change it.

To set the swap size:

- 1 Determine the current swap size for a node by issuing the command:

```
$ swap -l
```

You must run this command on each node of your system. The command displays information as on the following table:

swapfile	dev	swaplo	blocks	free
/dev/dsk/c0t3d0s1	32,25	8	132040	103440
/dev/dsk/c0t2d0s1	32,17	8	580600	553760
/dev/dsk/c0t1d0s6	32,14	8	1348192	1316864

A single *block* is 512 bytes. In this example, the total swap space is 2,060,832 blocks, or approximately 1GB.

- 2 If you are not already *root*, become *root*.
- 3 Change the swap space as necessary for each node, using the command:

```
swap -a.
```

Setting the Paging Space on Compaq Tru64

- 1 Determine the current swap size for a node by running the command:

```
$ /sbin/swapon -s
```

on each node of your system. The command displays information in the following format:

```
Swap partition /dev/rz0b (default swap):
  Allocated space:      786432 pages
  (6144MB)
  In-use space:         1 pages ( 0%)
  Free space:          786431 pages ( 99%)

Total swap allocation:
  Allocated space:     786432 pages (6144MB)
  Reserved space:      12110 pages ( 1%)
  In-use space:         1 pages ( 0%)
  Available space:     774322 pages ( 98%)
```

In this example, the total swap space is slightly less than 6 GB.

- 2 As *root*, add swap space as necessary for each node, using the command:

```
# swapon -a
```

Setting the Paging Space on HP-UX

- 1 Log on as *root*.
- 2 Determine the current swap size for a node by issuing the command:

```
# swapinfo
```

You must run this command on each node of your system. The command displays information in the form shown in the following table:

Type	KB Avail	KB Used	KB Free	% Used	Start/Limit	KB Reserve	Pri	Name
dev	524288	0	524288	0%	0		1	/dev/vg00/lvol2
dev	524288	60808	463480	12%	0		0	/dev/vg01/1vol1
reserve		240012	240012					
memory	697588	602088	95500	86%				

In this example, the total swap space for the two disks is approximately 1 GB.

Caution. Changing paging space requires root privileges. In general, only a system administrator should reset it. The procedure is potentially destructive.

- 3 Change the swap space as necessary for each node, using *sam* (System Administration Manager) or with */etc/swapon -a*.

Setting the Paging Space on LINUX

For a guide to setting up paging space on a Red Hat LINUX system, see <http://www.redhat.com/docs/manuals/linux/RHL-8.0-Manual/custom-guide/ch-swapspace.html>.

Edit /etc/magic

Note This procedure is optional.

The UNIX *file* command examines a file to classify it. The command reads the file */etc/magic* to identify files that have a *magic* number, that is, a constant indicating the file type.

The file *install_dir/etc/magic* contains additions to */etc/magic* that are specific to the parallel engine, where *install_dir* is the top-level directory of your parallel engine installation. Edit your system's */etc/magic* by appending the Parallel-Extender-specific additions to it. You must be logged in as root to edit */etc/magic*.

Here is a sample *file* command and sample system output after the edits described above have been made:

```
$ file t1.ds
t1.ds: Ascential Software parallel engine dataset descriptor
```

Setting Environment Variables

You need to set various environment variables to ensure smooth operation of your parallel engine. Environment variables are set on a per-project basis from the DataStage Administrator client, using the **Properties...** button on the Project Properties **General** page (see "[Setting Environment Variables](#)" in *DataStage Administrator Guide* for details).

Temporary Directory

By default, the parallel engine uses the directory */tmp* for some temporary file storage. If you do not want to use this directory, assign the path name to a different directory through the environment variable *TMPDIR*.

Network Settings

If your system connects to multiple processing nodes by means of a network, you need to set the *APT_IO_MAXIMUM_OUTSTANDING* environment variable to specify the amount of memory, in bytes, reserved for the parallel engine on every node for TCP communications. The default value is 2 MB.

If TCP throughput at that setting is so low that there is idle CPU time, increment it by doubling it until performance improves. If the system

is paging, however, or if your job fails with messages about broken pipes or broken TCP connections, the setting is probably too high.

Transform Library

If you are working on a non-NFS MPP system you need to set the `APT_COPY_TRANSFORM_OPERATOR` to true in order to enable Transformer stages to work in this environment. DataStage users must have the appropriate privileges to create project directory paths on all the remote nodes at runtime.

`APT_COPY_TRANSFORM_OPERATOR` is set false by default.

Job Monitoring

By default the job monitor uses time-based monitoring in the DataStage Director, i.e., the job monitor window is updated every five seconds. But you can also specify that the monitoring is based on size, i.e. the job monitor window is updated every so many entries. Do this by setting a value for the `APT_MONITOR_SIZE` environment variable. If you override the default setting for the `APT_MONITOR_TIME` (i.e., set it to something other than 5), this will in turn override `APT_MONITOR_SIZE`.

Setting Transport Block Size

You can set the `APT_AUTO_TRANSPORT_BLOCK_SIZE` to True to have DataStage automatically calculate the block size for transferring data internally as jobs run.

The `APT_DEFAULT_TRANSPORT_BLOCK_SIZE` variable defines the block size used otherwise. By default it is set to 32768.

Dumping Information

Set `APT_DUMP_SCORE` to True to have the parallel engine report detailed information about jobs as they run. This is set to False by default.

C++ Compiler

You require a C++ compiler on your machine to use the parallel engine. You need the C++ compiler specific to your platform. See "[C++ Compilers and Run-Time Libraries](#)" on [page 3-13](#) for details.

The environment variables `APT_COMPILER` and `APT_LINKER` are set at install to point to the default locations of these compilers. If your compiler is installed somewhere else, you will need to change the default environment variables for every project using the DataStage Administrator. See "[Setting Environment Variables](#)" in *DataStage Administrator Guide*.

Configuring the Parallel Environment

For you to develop and run parallel jobs, your development directory must be globally visible to all other nodes in the system. This often means the directory must be globally cross-mounted.

To configure the parallel environment:

- 1 Make your development directory visible to all other nodes in the system with the same path name. Typically, you use NFS to make the directory globally visible.
- 2 If you have a network-cluster system, you are using the Bourne or C shell, and your development directory is not statically mounted or automounted, set the environment variable `PWD` to the globally-visible path of the directory. The directory must be visible with the same path name on all processing nodes on which parallel jobs are run. Update `PWD` whenever you change to a new directory.

For example, if you use C shell and the directory `/usr/me/parallel-extender/app1` as your development directory, set `PWD` as shown here:

```
$ setenv PWD /usr/me/parallel-extender/app1
```

After you have completed this configuration, you can copy the sample applications from the directory `install_dir/examples` to your directory and run the examples

Granting User Privileges to run Parallel Jobs

A user who runs a parallel job must have the necessary privileges on all nodes used by that application, including:

- Login access
- Read/write access to disk and scratch disk resources
- Execute permissions on local copies of programs and scripts
- Read/write permissions to the temporary directory

- Read access to `$APT_ORCHHOME`

Upgrading from Beta Releases

If you are upgrading from a Beta release of DataStage 7, you will have redundant copies of the parallel engine on your system. You can save yourself some space by manually deleting these redundant copies after you have upgraded.

To locate the redundant copies, go to your DataStage server installation directory (`$DSHOME/..`). List the directory and look for the symbolic link `PXEngine`. The directory this points to is the current parallel engine, it has a name in the form `PXEngine.700.n`, where `n` is the subversion number, and is located in the same directory (i.e., `$DSHOME/..`). You can safely delete all parallel engine directories with **different** subversion numbers.

Setting Up Database Connectivity

This section outlines the steps you need to take to connect to various databases from a parallel job. This information is also given in the relevant stage editor chapter in the [Parallel Job Developer's Guide](#).

DB2/UDB Databases

DataStage should be configured to run on all processing nodes functioning as DB2 servers. Do this by including node definitions for your DB2 server nodes in your configuration file (see "[The Parallel Engine Configuration File](#)" in [Parallel Job Developer's Guide](#) for details about the configuration file, and particularly the resource DB2 option).

Calling the DB2 Setup Script

In order to connect to DB2/UDB databases you need to run the script `$APT_ORCHHOME/bin/db2setup.sh` from the UNIX command line to configure DataStage to access the DB2 database. The file must be called once for each DB2 database to be accessed by DataStage Enterprise Edition users. Pass the database name as an argument.

For example, the following command calls `db2setup.sh` to configure DataStage to access the database `db2_8`:

```
$ db2setup.sh db2_8
```

Granting User Privileges

You must grant privileges to each user who will run jobs containing the DB2/UDB Enterprise stages by running the `$APT_ORCHHOME/bin/db2grant.sh` script once for each user. You need DBADM privileges to run this script. The syntax of the command is:

```
$ db2grant.sh database_name user_name
```

Remote Connection

You can also connect from a DB2/UDB Enterprise stage to a remote DB2 Server. The connection is made via a DB2 client. In order to remotely connect from a DB2 client to a DB2 server, the DB2 client should be located on the same machine as the DataStage server. Both DB2 client and DB2 server need to be configured for remote connection communication (see your DB2 Database Administrator).

The DataStage configuration file needs to contain the node on which DataStage and the DB2 client are installed and the nodes of the remote computer where the DB2 server are installed (see "[The Parallel Engine Configuration File](#)" in *Parallel Job Developer's Guide* for information about the configuration file and how to set it up).

On the DB2/UDB Enterprise stage in your parallel job, you need to set the following properties:

- **Client Instance Name.** Set this to the DB2 client instance name. If you set this property, DataStage assumes you require remote connection.
- **Server.** Optionally set this to the instance name of the DB2 server. Otherwise use the DB2 environment variable, DB2INSTANCE, to identify the instance name of the DB2 server.
- **Client Alias DB Name.** Set this to the DB2 client's alias database name for the remote DB2 server database. This is required only if the client's alias is different from the actual name of the remote server database.
- **Database.** Optionally set this to the remote server database name. Otherwise use the environment variables APT_DBNAME or APT_DB2DBDFT to identify the database.
- **User.** Enter the user name for connecting to DB2, this is required for a remote connection.
- **Password.** Enter the password for connecting to DB2, this is required for a remote connection.

You can use DataStage's remote connection facilities to connect to different DB2 server within the same job. You could, for example, read from a DB2 database on one server, use this data to access a lookup

table on another DB2 server, then write any rejected rows to a third DB2 server. Each database would be accessed by a different stage in the job with the Client Instance Name and Server properties set appropriately.

Performing System Configuration of DB2

DB2 can be configured in many ways on your system. The DB2 administrator typically follows this procedure to configure DB2 and DataStage to communicate with one another:

- 1 Install DB2 and make sure it is working properly.
- 2 Grant the DataStage users SELECT privileges on the system tables `syscat.nodegroupdef`, `syscat.tablespace`, and `syscat.tables`.
- 3 Make the file `db2nodes.cfg` readable by the DataStage administrative user.
- 4 DataStage runs many processes for each job. This can require the system administrator to modify DB2 resources, configuration parameters, and manager configuration parameters of your system. See the DB2 administration manuals for more information.
- 5 The DB2/UDB Enterprise stage requires that users invoking it in load mode have DBADM privilege on the DB2 database written to by the stage. Among the ways to grant this privilege is to start DB2, connect to a database, and grant privileges as follows:

```
db2> CONNECT TO database_name
db2> GRANT DBADM ON DATABASE TO USER user_name
```

Performing User Configuration of DB2

To access DB2, DataStage parallel jobs connect to DB2 using an AIX user name. Therefore, DataStage users who invoke DB2 read operators must have SELECT privilege on the table they access, and users who invoke DB2 write operators must have INSERT privileges on the tables they access.

The following procedure configures an Orchestrate user to access DB2. This procedure can be performed by either a system administrator or a DataStage user.

- 1 Set the environment variable DB2INSTANCE to the user name of the owner of the DB2 instance. DB2INSTANCE determines the location of `db2nodes.cfg`, as follows:

```
~owner_user_name/sql1ib/db2nodes.cfg
```

For example, if you set DB2INSTANCE to Jan, the location of `db2nodes.cfg` is

```
~Jan/sqllib/db2nodes.cfg.
```

The file *db2nodes.cfg* contains information used to translate DB2 node numbers to host names. The information in this file must agree with the host names specified in the DataStage configuration file.

- 2 Optionally, set the environment variable DB2DBDFT to the name of the DB2 database that you want to use.
- 3 You must source the file *db2profile* and update your LIBPATH environment variable to include the location of the DB2 library containing your instance of DB2.

In the following example, the owner of the DB2 instance is the user *Mary*, the location of *db2profile* is *~Mary/sqllib/db2profile*, and the location of the DB2 library is *~Mary/sqllib/lib*.

For Korn and Bourne shells, include the following statements in your *.profile* or *.kshrc*:

```
. ~Mary/sqllib/db2profile
export LIBPATH=~Mary/sqllib/lib:${LIBPATH:~/usr/lib}
```

For C shell, put these lines in your *.cshrc*:

```
source ~Mary/sqllib/db2profile
if (! $?LIBPATH) setenv LIBPATH /usr/lib
setenv LIBPATH ~Mary/sqllib/lib:$LIBPATH
```

Oracle Databases

Accessing the Oracle Parallel Server

This is optional and applies to the AIX and Tru64 platforms only. To access the Oracle Parallel Server, users must have SELECT access to the *sys.gv_\$instance* and *sys.v_\$cache* tables. Issue the following SQL statements to grant this access:

```
GRANT select ON sys.gv_$instance TO public;
GRANT select ON sys.v_$cache TO public;
```

If you do not have Oracle OPS on these platforms, you should set the *APT_ORACLE_NO_OPS* environment variable to disable OPS checking on the Oracle Enterprise stage.

Oracle System Administration

You need to be running Oracle 8 or better, Enterprise Edition in order to run DataStage jobs using the Oracle Enterprise Stage. Your Oracle installation should include the Oracle Database Utilities and Oracle Network software.

You must also do the following:

- 1** Create the user defined environment variable `ORACLE_HOME` and set this to the `$ORACLE_HOME` path (e.g., `/disk3/oracle9i`)
- 2** Create the user defined environment variable `ORACLE_SID` and set this to the correct service name (e.g., `ODBCSOL`).
- 3** Add `ORACLE_HOME/bin` to your `PATH` and `ORACLE_HOME/lib` to your `LIBPATH`, `LD_LIBRARY_PATH`, or `SHLIB_PATH`.
- 4** Have login privileges to Oracle using a valid Oracle user name and corresponding password. These must be recognized by Oracle before you attempt to access it.
- 5** Have `SELECT` privilege on:
 - `DBA_EXTENTS`
 - `DBA_DATA_FILES`
 - `DBA_TAB_PARTITONS`
 - `DBA_TAB_SUBPARTITIONS`
 - `DBA_OBJECTS`
 - `ALL_PART_INDEXES`
 - `ALL_PART_TABLES`
 - `ALL_INDEXES`
 - `SYS.GV_$INSTANCE` (Only if Oracle Parallel Server is used)

Note `APT_ORCHHOME/bin` must appear before `ORACLE_HOME/bin` in your `PATH`.

We suggest that you create a role that has the appropriate `SELECT` privileges, as follows:

```
CREATE ROLE DSXE;
GRANT SELECT on sys.dba_extents to DSXE;
GRANT SELECT on sys.dba_data_files to DSXE;
GRANT SELECT on sys.dba_tab_partitions to DSXE;
GRANT SELECT on sys.dba_tab_subpartitions to DSXE;
GRANT SELECT on sys.dba_objects to DSXE;
GRANT SELECT on sys.all_part_indexes to DSXE;
GRANT SELECT on sys.all_part_tables to DSXE;
GRANT SELECT on sys.all_indexes to DSXE;
```

Once the role is created, grant it to users who will run DataStage jobs, as follows:

```
GRANT DSXE to <oracle userid>;
```

Teradata Databases

Installing the Teradata Utilities Foundation

You must install Teradata Utilities Foundation on all nodes that will run DataStage parallel jobs. Refer to the installation instructions supplied by Teradata. (You need system administrator status for the install.)

Creating Teradata User

You must set up a Teradata database user (this is the user that will be referred to by the DB options property in the Teradata stage). The user must be able to create tables and insert and delete data. The database for which you create this account requires at least 100 MB of PERM space and 10 MB of SPOOL. Larger allocations may be required if you run large and complex jobs. (You need database administrator status in order to create user and database.)

The example below shows you how to create the orchserver account. The user information is stored in the terasync table. The name of the database in this example is userspace. The following four commands for BTEQ set up the account:

```
CREATE USER orchserver FROM userspace AS
PASSWORD = orchserver
PERM = 100000000
SPOOL = 10000000
```

Once the account is set up, issue the following command:

```
GRANT select ON dbc TO orchserver;
```

Creating a Database Server

If you want to use a pre-existing Teradata user, you only need install a database server and configure it to use a new database. Install the new database server with the same PERM and SPOOL values as shown above. Here is an example of creating a database server called devserver using table userspace:

```
CREATE DATASBASE devserver FROM userspace AS
PERM = 100000000
SPOOL = 10000000
GRANT create table, insert, delete, select ON devserver TO orchclient;
GRANT create table, insert, delete, select ON devserver TO orchserver;
```

Informix XPS Databases

You must have the correct privileges and settings in order to use the Informix Enterprise Stage. You must have a valid account and appropriate privileges on the databases to which you connect.

You require read and write privileges on any table to which you connect, and Resource privileges for using the Partition Table property on an output link or using create and replace modes on an input link.

To configure access to Informix XPS:

- 1 Make sure that Informix XPS is running.
- 2 Make sure the INFORMIXSERVER is set in your environment. This corresponds to a server name in sqlhosts and is set to the coserver name of coserver 1. The coserver must be accessible from the node on which you invoke your DataStage job.
- 3 Make sure that INFORMIXDIR points to the installation directory of your INFORMIX server.

Make sure that INFORMIXSQLHOSTS points to the sql hosts path (e.g., `/disk6/informix/informix_runtime/etc/sqlhosts`).

Using Informix XPS Stages on AIX Systems

In order to run jobs containing Informix XMS stages on AIX systems, you need to have the Informix client sdk 2.81 version installed along with the Informix XPS server. The LIBPATH order should be set as follows:

```
LIBPATH=$APT_ORCHHOME/lib:$INFORMIXDIR/lib:`dirname $DSHOME`/  
branded_odbc/lib:$DSHOME/lib:$DSHOME/uvd11s:$DSHOME/java/jre/bin/  
classic:$DSHOME/java/jre/bin:$INFORMIXDIR/lib:$INFORMIXDIR/lib/  
cli:$INFORMIXDIR/lib/esql
```

Installing DataStage Components on a Mainframe Platform

MVS

If you are installing Enterprise MVS Edition (see "[Enterprise MVS Edition](#)" on [page 2-2](#)), there are various components that you need to install on the mainframe platform where you will be running your DataStage jobs. (If you are not installing Enterprise MVS Edition you can skip this chapter.)

The Directory Mainframe Components on the DataStage CD contains the DataStage mainframe components:

- **ArdtMsg1.cob.** A COBOL copy library file.
- **ArdtMsg2.cob.** A COBOL copy library file.
- **ArdtUBgn.cob.** A COBOL copy library file.
- **ArdtUCod.cob.** A COBOL copy library file.
- **ArdtUDat.cob.** A COBOL copy library file.
- **ArdtUEnd.cob.** A COBOL copy library file.
- **RTLMsgs.cob.** A COBOL copy library file.
- **ArdtIMsg.cob.** A COBOL copy library file.
- **ArdtRTL.data.** The DataStage runtime library, in FTP format.
- **ArdtRTL.JCL.** Sample JCL to install the runtime library.
- **Readme.txt.** Description of the installation procedure.

Installing the Copylib Members

To install the copylib members:

- 1 Add ArdtMsg1.cob to the mainframe copy library as member name ARDTMSG1.
- 2 Add ArdtMsg2.cob as ARDTMSG2.
- 3 Add ArdtUBgn.cob as ARDTUBGN.
- 4 Add ArdtUCod.cob as ARDTUCOD.
- 5 Add ArdtUDat.cob as ARDTUDAT.
- 6 Add ArdtUEnd.cob as ARDTUEND.
- 7 Add RTLMsgs.cob as RTLMSGs.
- 8 Add ArdtIMsg.cob as ARDTIMSG.
- 9 In the DataStage Manager, edit the JCL templates for CompileLink and DB2CompileLinkBind. In the COBCOMP step, make sure that the copy library you used above is in the concatenation of libraries for the SYSLIB DD statement.

Installing the Runtime Library (RTL)

To install the run time library, you must:

- Allocate an RTL data file on the mainframe
 - Transfer the RTL data file to the mainframe
 - Run the TSO Receive command
 - Transfer the IEBCOPY JCL to the mainframe
 - Run the IEBCOPY job
 - Edit the JCL templates
- 1 Allocate an RTL data file on the mainframe.

Before you transfer the data file, you must log on to your mainframe and pre-allocate the dataset to which the data file is sent. To do this, use ISPF function 3.2 (Data Set Utility). Request the Allocate a new dataset function, then enter a dataset name. On the subsequent ISPF screen, enter the following data:

```
Space units:TRACK (or TRKS)
Primary quantity:10
Secondary quantity:10
Directory blocks: 0
Record format:FB
Record length:80
Block size:3120
```

2 Transfer the RTL data file to the mainframe.

You must transfer the RTL data file to the data file on the mainframe using FTP.

- a** From your workstation, start FTP.
- b** Connect and signon to the mainframe.
- c** Enter the BIN command to specify binary transfer mode.
- d** Run the PUT command to send file ArdtRTL.data to the mainframe dataset you allocated in step 1.

3 Run the TSO RECEIVE command:

- a** Log on to TSO on the mainframe.
- b** Execute the command:

```
RECEIVE INDA('dataset_name')
```

dataset_name is the name of the dataset allocated in step 1.
This prompt appears:

```
Enter restore parameters or 'DELETE' or 'END' +
```

- c** In response to the prompt, enter:

```
DA('filename')
```

filename is a name of your choice. This file is created by the RECEIVE command.

4 Transfer the IEBCOPY JCL file to the mainframe:

- a** From your workstation, start FTP.
- b** Connect and signon to the mainframe.
- c** Run the PUT command to send file ArdtRTL.jcl to the mainframe dataset.

5 Run the IEBCOPY job:

- a** Edit the JCL file created on the mainframe in step 4, to customize the JCL for your site:

Change the JOB card per you site standards.

Change the DSN on the SEQ1 card to be the name of the dataset created by the RECEIVE command in step 3.

Change the DSN on the PDS2 card to be the name of a link library that is used to contain the runtime library module.

- b** Submit the job.

- 6 Edit the JCL templates. In the DataStage Manager, edit the JCL templates for CompileLink and DB2CompileLinkBind. In the LKED step, change the DSN to be the dataset name you used on the PDS2 card in step 5a.

Running the Mainframe IVP

Directory Utilities\Supported\IVP on the DataStage CD contains the Enterprise MVS Edition Installation Verification Process (IVP). The IVP consists of two jobs that are designed to validate that your physical environment is properly configured prior to using mainframe jobs. After DataStage installation is complete, refer to the readme file in this directory for detailed instructions on installing and running the IVP.

Installing DataStage Components on a USS System

If you are intending to run DataStage parallel jobs on USS (z/OS) systems, then you need to install the parallel engine on the USS machine. (If you do not intend to do this you can ignore this chapter.)

The USS components are copied the DataStage server from a dedicated USS CD. You need to copy all the files from the CD to the server. These components need to be FTPed from your DataStage server machine to your USS machine. We recommend that you TAR the files before transferring them, so that the correct file permissions are preserved. You can FTP to any location on the USS machine. The actual install is performed by running the script *install* from that location.

Requirements

The basic requirements of your setup are as follows:

System Requirements

Your USS machine must meet the following system requirements:

- Minimum 4 CPU machine (600 MIPs+).
- z/OS 1.3 or greater.
- IBM's C/C++ compiler 1.2 (current version as of z/OS 1.2). This is required for jobs containing Transformer stages and custom build stages.
- Java 1.3 or better.

- rsh daemon running (when deploying under control of DataStage).

Hardware Requirements

To install the parallel engine, your USS system should meet the following hardware requirements:

- 250 MB of free disk space for product installation.
- At least 128 MB of memory per processing node. 256 MB or more are recommended, depending on your application.
- At least 500 MB of scratch disk space per processing node.

You can install anywhere on your USS machine, the top-level directory is subsequently identified by the environment variable `APT_ORCHHOME` on the server machine.

Development Environment

To develop DataStage parallel jobs for deployment on a USS system, you require:

- DataStage clients on Windows platform
- DataStage server on UNIX platform

Connectivity

DataStage jobs run on your USS system will have the following connectivity:

- USS (HFS/ZFS) files (read and write files natively)
- MVS Datasets (read files natively)
- VSAM Datasets (read ESDS, KSDS, RRDS files natively)
- DB2 7.1 or 8.1
- Teradata 6.11 or 7.0

Running the Install

We assume you have already created an installation directory and FTPed the required files to it.

- 1 Log on to the system.
- 2 CD to the directory where you FTPed the components
- 3 Issue the following command to run the installation script:

```
./install
```

The installation procedure is automatically logged in */tmp/orchinstlognnn*, where *nnn* is the process ID of the installation script. This directory and file are volatile. To save the file permanently, move it elsewhere.

At any prompt you may give one of three special responses:

- Type a question mark (?) followed by a newline for help in replying to the prompt.
 - Type an exclamation mark (!) followed by a newline to start an interactive UNIX shell. To leave the shell, either press Ctrl-D or type exit followed by a newline to resume the installation.
 - Type quit! to exit the installation procedure.
 - Most prompts have defaults, which are enclosed in square brackets ([]). Enter a newline to accept the default.
- 4 The installation script prompts you for the name of your installation directory. By default, the installation directory is the value of the environment variable `APT_ORCHHOME`, if set. If `APT_ORCHHOME` is not set, the default is `/<userid>/apt` on the node to which you are logged in. If you specify a directory that does not exist, you are given a chance to create the directory. You must have write permission to the installation directory. If you do not, you are notified of the problem and asked again to supply an installation directory.
 - 5 The script prompts you for a locale choice. Your choice determines the language in which messages and exceptions are expressed. You can choose either English or Japanese. The default is English.

At this point, the installation program announces that it is installing several parallel engine components.

- 6 You are prompted to choose two TCP ports for use by the Job Monitoring Tool. You must pick ports that are not in use by any other service. The installation program offers two unused ports as defaults. You should register the ports in `/etc/services` or in any other service registry used by the system. The installation program reminds you to create and test a configuration file. It tells you to rename `$APT_ORCHHOME/etc/distribute-component.example` to `$APT_ORCHHOME/etc/distribute-component` if you wish to use custom operators.

The installation program is now complete.

Configuring the System

"*The Parallel Engine Configuration File*" in *Parallel Job Developer's Guide* describes the configuration files that specify what processing, storage, and sorting facilities on your system should be used to run a parallel job. You need to set up a configuration file on your USS system.

Remote Shell Capability

If you are intending to run parallel jobs on the USS machine under the control of the DataStage Director, then the parallel engine must be able to run the remote shell command *rsh* without a password on all processing nodes running parallel jobs. *rsh* is not part of a standard USS install, but the RSHD server is available for (free) download from IBM. Once installed, you need to start the *rsh* daemon from InetD.

The RSHD server provides server functions in the UNIX System Services environment for remote shell clients based on the remote shell (RSH) protocol. Commands are executed as UNIX System Services shell commands.

The server is installed into both the hierarchical file system and OS/390 V2R5 IP and later product libraries. In most installations, the module is installed into the */usr/lpp/tcpip/sbin* directory with the sticky bit, and named *orshd*. The corresponding symbolic link is created in */usr/sbin* directory. However, since it has the sticky bit, the executable module is loaded from the SEZALINK library data set. Note that there are options to change the mount point for the TCP/IP filesystem in a hierarchical file system environment, so the module *orshd* might be installed in another directory.

The default port number for *rsh* is 514, and this must be reserved in the PROFILE data sets of the TCP/IP stacks that act as UNIX system services AF_INET transport providers:

```
PORT
  514    TCP    OMVS;REMOTE SHELL SERVER
```

If the RSH server is the first server controlled by InetD, you have to create an InetD configuration file, named */etc/inetd.conf*. The sample file is stored in */samples* directory, but you have to modify it to meet your UNIX System Services environment. In particular, you have to modify *inetd.conf* to point to */usr/sbin/orshd*.

Specifying the Location of *rsh*

To find *rsh* on a processing node, the parallel engine searches the following paths in the order given:

```
install_dir/etc/remsh (if it exists)
/usr/ucb/rsh
/usr/bin/remsh
/bin/remsh
/usr/bin/rsh
```

where *install_dir* is the top-level directory of your parallel engine installation.

If the parallel engine does not find your *rsh* command, you must specify its location. To do so, copy or rename the supplied file *install_dir/etc/remsh.example* to *install_dir/etc/remsh*. This file contains the following shell script:

```
#!/bin/sh
# Example apt/etc/remsh
exec /usr/bin/rsh "$@"
```

As written, this shell script invokes */usr/bin/rsh*. Edit the last line of this script to invoke your specific remote shell command. The script should be executable by all users. Use *chmod* to ensure that it is:

```
# chmod 755 script-filename
```

Test this by running *rsh* on each node:

```
# rsh nodename uptime
```

Allowing User Execution of *rsh* Without a Password

In order to execute *rsh* on the USS machine without a password, you have to set up a user exit called */usr/sbin/ruserok* in order to enable the USS to recognise the remote machine as a trusted host. The user exit returns 0 if it recognises the host, and the command is executed. Otherwise the connection will be terminated.

The following is an example user exit:

```

/*****
/* ruserok.c This is an example RSHD installation exit. Place */
/* executable in /usr/sbin/ruserok. */
/*****

#include <stdio.h>
#include <string.h>
#include <fcntl.h>

int main(int argc, char** argv)
{
    char *rhost1; /* "hostname" or "hostname.domain" of client
```

```

        obtained by caller:
        gethostbyaddr(getpeername()) */
int cliuid; /* uid of user name on this systems */
char *cliuname; /* user name on client's system */
char *servuname; /* user name on this (server's) system */
        /* the list of trusted users */
char *u_trusted[] = { "woza", "gdente", "silviar",
                    "eikens", "kakky" };

int cnt;

rhost1 = *(argv+1);
cliuid = atoi(*(argv+2));
cliuname = *(argv+3);
servuname = *(argv+4);

{ /* write logging information to a file. */
    int fdes;
    char buf[80];
    memset(buf, '\0', sizeof(buf));
    fdes=open("/tmp/rulog", O_WRONLY | O_CREAT | O_APPEND, 0666);
    if(fdes!=EOF){
        sprintf(buf, "rhosts = %s, uid=%d, cliname=%s,
                    srvname=%s\n", rhost1, cliuid, cliuname,
                    servuname);

        write(fdes, buf, strlen(buf));
        close(fdes);
    }
}
    /* check if the userid is trusted. */
for(cnt=0; cnt<5; cnt++){
    if( (strcmp(servuname, *(u_trusted+cnt))==0))
        return(0); /* the userid is trusted. */
}
return(1); /* reject this client */
}

```

Setting Environment Variables

You need to set various environment variables to ensure smooth operation of your parallel engine. These are set on the DataStage server machine, using the DataStage Administrator.

Location of Parallel Engine

Set the APT_ORCHHOME environment variable to specify the location of the parallel engine on the USS system.

Location of Configuration File

You should set `APT_CONFIG_FILE` to identify the configuration file used to run jobs on the USS system (you cannot use the default configuration file supplied with DataStage for this). You can edit the file on your DataStage server, and it will be automatically copied to the USS system when you save it. You can either set `APT_CONFIG_FILE` to identify the configuration file on the DataStage server or to give its location on the USS system.

Temporary Directory

By default, the parallel engine uses the directory `/tmp` for some temporary file storage. If you do not want to use this directory, assign the path name to a different directory through the environment variable `TMPDIR`.

Job Monitoring

Only size-based monitoring is available on jobs run on a USS system. This means that the job monitor window is updated every so many entries. Specify the number of entries by setting a value for the `APT_MONITOR_SIZE` environment variable.

Setting Transport Block Size

You can set the `APT_AUTO_TRANSPORT_BLOCK_SIZE` to `True` to have DataStage automatically calculate the block size for transferring data internally as jobs run.

The `APT_DEFAULT_TRANSPORT_BLOCK_SIZE` variable defines the block size used otherwise. By default it is set to 32768.

C++ Compiler

You require a C++ compiler on your USS machine to use the parallel engine. You need the C++ compiler specific to your platform.

The environment variables `APT_COMPILER` and `APT_LINKER` are set at install to point to the default locations of these compilers. If your compiler is installed somewhere else, you will need to change the default environment variables.

Setting Up DB2 Access

Before DB2 can be used in a USS DataStage job, the parallel engine must be bound to a plan in DB2. This is done by using two files that were copied from the CD: *BindDB2.jcl* and *db2esql.bnd*.

BindDB2.jcl is a file that contains MVS JCL statements. These JCL statements define an MVS job that will bind a plan named "DB2ESQL" to DB2. The other file, *db2esql.bnd*, is used by the job.

The JCL statements must first be customized and put into compliance with the JCL standards of the customer site. A person knowledgeable in these standards will need to make these changes. All the strings in the file that begin with the "%" character need to be replaced with site-valid values. In addition, lines marked with "<== REVIEW" will need to be reviewed to make sure the dataset names are valid.

Once the JCL has been customized to site standards, it needs to be copied (via FTP or other means) to a dataset on the mainframe. Usually, a mainframe user will have a TSO partitioned dataset (PDS) that contains JCL member files. The *BindDB2.jcl* file needs to be copied to a member in the PDS. When that has been done, a person authorized to submit DB2 bind jobs must submit the job for execution.

After the job successfully runs, the DataStage parallel engine can access the DB2 database. Note that the bind step of this job needs to be run for each DB2 sub-system that is to be accessed by DataStage.

New Features in DataStage

This chapter describes features that are new to DataStage. You will find this of interest if you are upgrading an existing DataStage installation.

What's New at Release 7.5.1

Release 7.5.1 brings many new and improved features to DataStage.

SQL Builder

The DB2/UDB and Oracle stages now benefit from a new SQL Builder, available from the stage editor. This provides an easy-to-use interface for building queries to read Oracle and DB2 databases from server and parallel jobs.

Interaction with MetaStage

Various improvements have been made to the generation and handling of process meta data by DataStage jobs. These lead to a tighter integration between DataStage and MetaStage.

USS Support

DataStage now allows you to run parallel jobs on USS (z/OS) systems. Enhancements have been made to the Complex Flat File stage, which allows you to access mainframe data from within parallel jobs.

Remote Deployment

Remote deployment of parallel jobs allows job scripts to be stored and run on a separate machine from the DataStage Server machine. The remote deployment option can, for example, be used to run parallel jobs on a computer grid.

Enhancements to Job Reporting

The HTML job report available from the DataStage Designer now allows a user to hotlink from the job image to the job properties. A set of extra stylesheets are available on the DataStage installation CD in the folder Utilities\Unsupported\Stylesheet.

Multi-job Compiler

The compiler wizard now allows you to compile server job routines and any parallel build stages (buildops) that you have defined.

Enhancements to Command Line and API Facilities

New features for searching for jobs are now available both from the command line and from the API. Administration features previously only available in the DataStage Administrator client are now also available from the command line and the API.

Specifying Icons for Custom Stages

You can now specify your own icons for any custom stages that you develop for parallel jobs.

Vector Support in Parallel Transformer Stages

You can now access vectors within the parallel job Transformer stage.

Editing of Multiple Derivations Columns

You can now edit multiple derivations in column definitions where you are aggregating data in a certain stage editors (this facility was previously only available in the Transformer stage).

What's New at Release 7.5

Release 7.5 brings many new and improved features to DataStage.

Job Sequencer

Manny enhancements have been made to the job sequencer, including the ability to loop within a sequence, the option to automatically handle errors, and the capability of restarting a sequence from a checkpoint position.

Mobile Director

A light-weight Director client, called the Mobile Director, is available for use on WinCE-based poscket PCs. This allows you to run and monitor DataStage jobs while on the move.

USS Support

DataStage now allows you to run parallel jobs on USS (z/OS) systems. A new Complex Flat File stage allows you to access mainframe data from within parallel jobs.

Improved Designer Interface

The Designer now gives you visual cues telling you where stages are incomplete, and what information needs to be added.

Client Licenses

You now have the ability to update client licenses without needing to reinstall.

Job Reporting

HTML-based job reporting is now available from the DataStage Designer.

Multi-Client Manager

You can now maintain several different version of the DataStage clients on the same machine, allowing you to switch between them and connect to different servers.

Visual Cues

Visual cues are now available in parallel jobs, job sequences and shared containers that tell you where jobs will fail to compile, and what information is needed to remedy this.

Parallel Job Message Handlers

Gives you the ability to specify how certain log message types are handled, making for an easier-to-read job log.

Parallel Job Lookup Stage

The parallel job Lookup stage now has a new, improved user interface, making it easier to use.

Previous Enhancements

This section describes some previous enhancements that are important if you are upgrading from a pre-release 5.1 version of DataStage. If you are upgrading from Release 5.1 or better, you can skip this section.

New DataStage Server

DataStage Version 5.1 introduced the new DataStage server. At previous releases this functionality was provided by UniVerse. A number of differences between the DataStage server and UniVerse may affect existing DataStage jobs. The following sections describe these differences:

- Changes from UniVerse
- Name changes
- Loopback functionality
- UniVerse stage

Changes from UniVerse

The following functionality has changed for the DataStage server:

- BASIC object code compatibility
- UV/NET
- UV/ODBC
- Spooler
- Transaction logging
- UCI
- Replication

BASIC Object Code Compatibility

The DataStage server only supports object code produced by the DataStage server and Version 9.6 and earlier of UniVerse.

This means that BASIC code that is compiled by the DataStage server does not run in any version of UniVerse. You must recompile it in UniVerse.

Any code compiled in a version later than 9.6 of UniVerse does not run on the DataStage server without recompilation.

It is possible that code from UniVerse may not compile in DataStage server and vice versa at future releases of each product.

UV/NET

The UV/Net subsystem allows remote access of UniVerse files over a network between two UniVerse systems. One system is the server and has the file residing on it. The other system is the client that wants to access the file on the server as it does any local file. Both machines can use the same file at the same time, that is, they each honor the locks of the other and see the changes made by the other.

While this system has been used by DataStage to access UniVerse files on remote systems, UniVerse or DataStage systems do not access DataStage files in this manner. This means that only the client part of UV/Net is required for DataStage, as it only needs to communicate with a server and not handle incoming requests from a client.

The server part of UV/Net has been removed from DataStage. This means that it is not possible to have DataStage 7.5.1 act as a server of files to either UniVerse or another DataStage system. DataStage 6.0 or later still has the ability to access remote files on UniVerse systems. However, it cannot handle a request to access its files from a remote UniVerse or DataStage system.

UV/ODBC

DataStage does not use UV/ODBC, so this functionality has been removed. It should be noted, however, that UV/ODBC can still be used as a data source like any other ODBC driver, but it is no longer shipped as part of the DataStage product. It is only required by a UniVerse user who wants to use ODBC to communicate with UniVerse.

This means that ODBC clients cannot make connections to the DS 7.5.1 server. ODBC cannot be used to communicate with the DataStage server and access its files.

Spooler

In a UniVerse installation all printing and spooling is done from UniVerse through the spooler. Users do not print or spool any information from the DataStage server, only from the DataStage clients. The spooler is therefore redundant and has been disabled.

Transaction Logging

The transaction subsystem provides the ability to manage an update to a file as an atomic unit and to log information for that unit for replay in the future. The LOGGING of the transaction information is not used by DataStage as most of the data is transitory. The LOGGING capability for transaction support has been removed from DataStage.

As well as the logging of transactions, the transaction subsystem also provides support for warm start recovery. This lets a system recover a file back to the point where the system crashed or failed. Warm start recovery requires the LOGGING information to recover a file to a known state. Since the LOGGING feature has been removed, warm start recovery has also been removed from DataStage.

Note The normal use of transactions in programs is unaffected.

UCI

UCI is a SQL-based C API for UniVerse. It provides all the features required for SQL applications from the C language. DataStage does not use the UCI client and although DataStage does use BCI (the BASIC equivalent of UCI) to communicate with itself it does not use the UCI server as connections (all of which are local to the DataStage server) are done in process. The UCI has been removed from this release of DataStage.

This means that external programs that use the UCI to communicate with DataStage can no longer do so at this release.

Replication

The Replication subsystem has been removed from the DataStage server since it is not used by DataStage.

Name Changes

The new DataStage server contains new names for some existing programs. The leading *uv* is now a leading *ds*, for example:

Old Name	New Name
uvapi_slave	dsapi_slave

uvapi_server	dsapi_server
uvdlockd	dslockd

Loopback Functionality

Loopback functionality is now restricted. Previously you could configure the DataStage server *uvodbc.config* file so that a data source name pointed to the same machine on which DataStage was running. This functionality, where the host name is specified by its machine name or by its real IP address, no longer works. To access files in this way, you need to install a copy of UniVerse and configure the DataStage *uvodbc.config* file to point to that UniVerse.

Non-network loopbacks still work, that is, where the host name is specified as localhost or the special IP address 127.0.0.1 is used.

UniVerse Stage

Changes have been made to the behavior of the UniVerse stage. The UniVerse stage provides SQL-based connectivity to UniVerse databases and to tables held locally by the DataStage engine. This operates in two modes, which are best described as localuv and remote. Before Version 5.1, the behavior was as follows:

- **Localuv.** When used with a special data source name of localuv, the stage provided SQL connectivity to the local DataStage server. This mode of operation required no user configuration. It was typically used to gain access to extra functionality from UniVerse SQL that was unavailable from the hashed file stage. The UniVerse accounts accessible using localuv were the one or more DataStage project accounts **and possibly** other accounts that did not relate to DataStage projects (and had been explicitly created by a UniVerse-literate user). These non-DataStage accounts could exist at a DataStage installation without a separate installation of UniVerse.
- **Remote.** The stage can was used to connect to other remote instances of UniVerse. The *uvodbc.config* configuration text file in the project was edited as documented in "[Accessing UniVerse Systems](#)" in *DataStage Administrator Guide* to set up these connections. This could also be used to connect to other DataStage servers

DataStage 6.0 and later retains the existing localuv functionality for accessing DataStage project accounts, but you can no longer access non-DataStage accounts in this way. Such accounts are only maintained if you have chosen to set up a coresident UniVerse system. You connect to these as remote UniVerse accounts and need to take the following actions:

- Add an entry in the *uvodbc.config* file for the project to access the account using a remote connection
- Edit jobs to access the tables using this connection.

No changes are needed for jobs which use localuv to access files in DataStage projects.

There is no immediate change for UniVerse stages connecting to remote instances of the UniVerse product. The UniVerse stage can connect to all the remote UniVerse instances that it formerly did. However, this may change in the future if UniVerse is modified so that older versions can no longer connect to it. The DataStage server in Version 5.1.*n* is based on UniVerse 9.6.

Note You cannot use UniVerse stages to access data on remote DataStage systems that have upgraded to the Server engine (i.e. Release 5.1 onwards).

10

Troubleshooting

This chapter gives information about problems you might encounter installing or upgrading DataStage. It also has a section on problems and permanent restrictions.

Troubleshooting UNIX Installations

This section details problems you may encounter with DataStage running on UNIX platforms and suggests solutions.

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 *DSEngine* 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 as follows:

- 1 To stop the server:
`$DSHOME/bin/uv -admin -stop`
- 2 To upgrade configuration information for the server:
`$DSHOME/bin/uv -admin -regen`

- 3 To start the server:

```
$DSHOME/bin/uv -admin -start
```

Ensure that you allow sufficient time between executing stop and start commands (minimum of 30 seconds recommended).

Running Out of Memory on AIX

If you have jobs with large memory requirements and have been getting “unable to locate memory” errors on your AIX system, you can tune DataStage to increase your configured memory.

Take the following steps:

- 1 Log in as *dsadm* and edit the *uvconfig* file in the DataStage engine directory (for example, */u1/dsadm/AscentialDataStage/DSEngine/uvconfig*). Make the following changes:

- Change *DMEMOFF* to *0x90000000*
- Change *PMEMOFF* to *0xa0000000*

- 2 Ensure that there are no active DataStage users on the system, then shut down the DataStage server:

```
$DSHOME/bin/uv -admin -stop
```

- 3 To upgrade configuration information for the server, execute:

```
$DSHOME/bin/uv -admin -regen
```

- 4 Add the following line to the *dsenv* file (in the *dshome* directory):

```
LDR_CNTRL=MAXDATA=0x30000000;export LDR_CNTRL
```

- 5 Execute *dsenv* to apply the new environment settings.

- 6 Restart the DataStage server:

```
$DSHOME/bin/uv -admin -start
```

dsrpcd Issues

The DataStage RPC daemon, *dsrpcd*, 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*. For information about *dsrpcd* and its environment requirements, see ["Environment Variables"](#) on [page 3-28](#) and *DataStage Administrator Guide*. The *dsrpcd* daemon is started when you start the DataStage server.

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 -a | grep dsrpc
```

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. There can, however, be an exception to this if *dsapi_slave* or *dscs* processes are not cleaned up during DataStage shutdown. You should check for these processes and remove them if found.

ODBC Connection Problems

You should check that your connection details as specified in the *.odbc.ini* file are correct. Another Data Direct utility, called *demoodbc*, can be used to test the connection.

The utility is located in the directory `$DSHOME/./branded_odbc/demo`. It requires that you set up a table called EMP in your target database. Scripts to set up and populate EMP on common databases are provided in the *demo* directory. Once you have set up the database, run the utility as follows:

```
# demoodbc -uid <username> -pwd <password> <DSN>
```

Where *<username>* and *<password>* are the username and password required for connecting to the database and *<DSN>* is the data source name as specified in the *.odbc.ini* file.

If the connection fails you should check your entry for this data source in *.odbc.ini*.

(You can, if required, run *demoodbc* without setting up the EMP database, but you will need to examine the error message to determine if the program is connecting, but failing to find the EMP table, or whether it is failing to connect.)

Note The *demoodbc* utility is a Data Direct product. Ascential Software are not responsible for its maintenance or support

Problems Using Oracle 8 CLI Plug-ins

If you are having problems using the Oracle 8 CLI Plug-in, or the Oracle CLI bulk loader Plug-in, you should check the shared library linking requirements. Certain versions of Oracle 8 running on certain platforms require you to perform a one-time site linking and library rebuild. See the Technical Bulletins provided with the software for more details. These are accessible from the main PDF contents list

(open this by choosing **Ascential DataStage ► Online Manuals ► DataStage Documentation** from the Start Menu). They are in Acrobat format and require an Acrobat reader.

In some cases these problems might be obviated by installing an Oracle patch.

Login Problems on Windows Systems

If you have upgraded DataStage to the latest release, and previously logged in using the following scenario:

- The client and server are both on the same Windows server machine.
- You are using TCP/IP instead of lan pipes.
- You are not using a prefix in the username.
- You are logging in as dsadm, who is NOT an administrator.

You will find that this scenario no longer works. The solution is to make dsadm an administrative user.

Problems and Restrictions

The following sections contain information about problems and restrictions for Windows and UNIX platforms at this release:

- Stored Procedures Using ODBC - UNIX
- Connecting to the Server - UNIX
- Object Call Install - UNIX
- Deadlock Daemon - Windows and UNIX
- Environment Variables - UNIX
- Using ODBC stage to access Sybase on Compaq Tru64
- Using Oracle 8 OCI and ODBC stages in the same job

Stored Procedures Using ODBC - UNIX

Problems have been encountered with certain ODBC drivers properly reporting the number of columns in the result set when executing stored procedures, or attempting to retrieve meta data from stored procedures. These problems prevent DataStage from using these functions properly.

Connecting to the Server - UNIX

When connecting to a UNIX server from any DataStage client component such as the Designer or the Director, do not select the **Omit** option on the **Attach to Project** dialog box. The **Omit** option applies to connecting to DataStage Windows servers only.

Object Call Install - UNIX

Relevant object call files are now installed under the *ud41* directory at the same level in the file system as the DataStage server (for example, *\$DSHOME/./ud41*). Environment variables for object call are included in the standard *dsenv* file found in the DataStage server directory. Uncomment them to enable object call.

Deadlock Daemon - Windows and UNIX

The default state of the daemon is OFF. You must change this to ON in order to use the daemon. To do this, you must change the value of the *start* variable in the *dslockd.config* file in the DataStage server directory.

Using ODBC stage to access Sybase on Compaq Tru64

Jobs which use the ODBC stage to access Sybase on the Compaq Tru64 platform may encounter spurious errors giving a message like the following in the DataStage log file:

```
DataStage Job 19 Phantom 8139
Program "DSD.LinkReport": Line 257, Unable to open "DS_STAGETYPES" file.
access(DS_STAGETYPES/.Type1, F_OK) failed, errno = 0
DataStage Phantom Finished
```

This issue may be solved by upgrading to a later version of the Data Direct ODBC driver for Sybase on Compaq Tru64 (visit www.datadirect-technologies.com).

Using Oracle 8 OCI and ODBC Stages in the Same Job

You should not use Oracle 8 OCI and ODBC stages to access the same Oracle database in the same job. If you do the job may hang.



Upgrading Issues

This appendix contains information that is important if you are upgrading from releases of DataStage that are more than one version earlier than the new one. For this release, that means anything up to, and including, DataStage Release 7.1.

General Install Issues

This section describes installation issues that are common to both Windows and UNIX platforms.

Coresident UniVerse Systems – Windows Systems

From Release 5.1 onwards, DataStage uses the DataStage server engine rather than the UniVerse database to power it. If you are upgrading from DataStage 4.x or earlier, you are offered the option of preserving the UniVerse database as a coresident system during the server upgrade. All the DataStage projects and functionality will be moved to the upgraded version, but you can retain UniVerse as a standalone system. You would normally do this if you had non-DataStage accounts on the UniVerse system and wanted to maintain them. Otherwise you should chose to overwrite the existing installation. (See "[New DataStage Server](#)" on [page 9-4](#) for more information on the new server and coresident systems.)

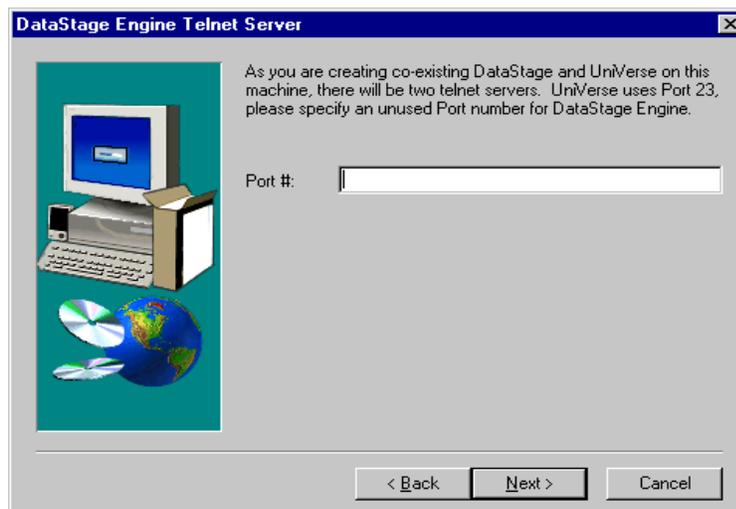
If you are upgrading from 5.1 or later, your existing installation is upgraded and you are not offered the option of setting up coresident systems.

If you are upgrading from 4.x or earlier, the setup program first asks if you want to preserve or overwrite the existing installation (see step 1 on [page 4-3](#)).

If you choose to upgrade the existing installation, the UniVerse database is completely removed from your system, and the new DataStage is installed into your existing DataStage directory structure.

If you choose to install DataStage Engine to a new location, the new DataStage engine will be installed in the Engine directory under the existing DataStage directory structure (for example `c:\Ardent\DataStage\Engine`), while the coresident UniVerse system is in the ServerEngine directory (for example `c:\Ardent\DataStage\ServerEngine`).

The setup program prompts you for a new port number for the new DataStage engine (UniVerse retains port 23). You can select any currently unused port. The `Winnt\System32\drivers\etc\Services` file gives a list of all currently used ports. Installing the DataStage server adds an entry called `dstelnet` which you can subsequently edit if required. If you are unsure about port numbers, ask your system administrator.



Once you have specified a new port number and chosen **Next**, the upgrade proceeds.

Coresident Systems – UNIX Systems

From Release 5.1 onwards DataStage uses the DataStage engine rather than UniVerse to power it. If you are upgrading from an earlier release, you have the option of installing the upgraded DataStage as a coresident system. This preserves the underlying UniVerse database. Accounts configured as DataStage projects are owned and managed

by the new DataStage installation. Accounts that are not configured as DataStage projects remain under the management of the existing installation, i.e., UniVerse. All references to DataStage projects or schemas in the previous installation are unloaded and deleted from the *UV.ACCOUNTS* file and SQL catalog. The unloaded data is subsequently reloaded into the new DataStage install prior to project upgrade. The actual DataStage project locations are not changed by this process.

You may want to have coresident systems if you use your existing UniVerse for other purposes besides running DataStage. Once you have upgraded you will no longer be able to access the coresident UniVerse as *localuv*, it becomes a remote UniVerse as if it was on a separate system.

If you maintain UniVerse as a coresident system you will have to specify a different install location for your DataStage, i.e. you cannot accept the default location. The new location can be a symbolic link. We recommend that you use the normal install location for DataStage systems `<dsadmhome>/Ascential/DataStage`.

For more information about the new server and its implications, see "[New DataStage Server](#)" on [page 9-4](#).

Verifying Coresident Installs

If you have chosen to retain the UniVerse database from your previous DataStage installation as a coresident system you should verify that both the upgraded DataStage and the coresident UniVerse are operational.

Certain key files and directories relating to the previous DataStage installation (besides the project directories) are copied to a new subdirectory `_dscoexist_` and deleted from the UniVerse directory. The fields and directories are as follows:

New Installation	Previous Installation
<code>_dscoexist_/bin</code>	<i>bin</i> directory (deleted DataStage binaries)
<code>_dscoexist_/lib</code>	<i>lib</i> directory (deleted DataStage libraries)
<code>_dscoexist_/JCLTemplates</code>	Mainframe JCL templates directory (if applicable)
<code>_dscoexist_/Template</code>	New project template directory
<code>_dscoexist_/Template.ini</code>	Template control information

New Installation	Previous Installation
<code>_dscoexist_/Plugins.ini</code>	Plug-in stage installation information
<code>_dscoexist_/dsenv</code>	Environment file used by <i>dsrpcd</i> daemon
<code>_dscoexist_/uv_account.u</code>	Unloaded UV.ACCOUNT information (DataStage accounts)
<code>_dscoexist_/uv_assoc.u</code>	Unloaded SQL catalog information (associations)
<code>_dscoexist_/uv_columns.u</code>	Unloaded SQL catalog information (columns)
<code>_dscoexist_/uv_schema.u</code>	Unloaded SQL catalog information (schemas)
<code>_dscoexist_/uv_tables.u</code>	Unloaded SQL catalog information (tables)
<code>_dscoexist_/uv_users.u</code>	Unloaded SQL catalog information (users)
<code>_dscoexist_/uv_views.u</code>	Unloaded SQL catalog information (views)

After you have verified that all projects and jobs have been successfully updated, you can delete the `_dscoexist_` directory.

XMLPack and ClickPack

Users of the existing XMLPack 1.0 and/or ClickPack must install the latest versions of these packages when upgrading from pre-Release 7.0 versions. The new XML Pack 2.0 is installed automatically. ClickPack is an add-on package.

Existing DataStage jobs using the XML Pack 1.0 will continue to work with the previous XML Pack. However, to take advantage of the new capabilities of XML Pack 2.0, existing applications should be re-designed using the stages in the new XML Pack. All new development going forward should use XML Pack 2.0.

XML Pack 2.0 and higher is not backward compatible with the previous version of XML Pack 1.0. Consequently, table definitions created by the previous version of the XML Pack cannot be reused with the new one. New table definitions must be created using the new XML Meta Data Importer (installed by default). No automatic migration is provided between the XML Pack 1.0 and XML Pack 2.0 as

each pack supports a different type of XML meta data. The 1.0 Pack supports XML DTD's which are strictly an XML legacy definition while the 2.0 pack supports XSD's. Consequently, we recommend you use a third party tool, e.g. XMLSpy, to generate an XSD compliant definition from your old DTD. Both XML Pack 1.0 and 2.0 can be used simultaneously in the same job. Existing jobs built with the XML Pack 1.0 can still be used without any modifications but require the installation of the XML Pack 1.0.

If you are upgrading from release 7.0 or 7.1, you must uninstall your current XML Pack client before upgrading DataStage. To do this:

- 1 Got to Windows control panel.
- 2 Choose Add or remove programs.
- 3 Select DataStage XML Pack and click **Remove**.

Oracle OCI 9 Plug-in Renamed

The Oracle OCI 9 plug-in has been renamed Oracle OCI since it is now also compatible with Oracle 10g. Because the plug-in's name is used to determine whether it has been previously installed in an earlier version of DataStage, the DataStage installer will not recognize if the plug-in is being upgraded and thus will not preselect the plug-in for installation. To upgrade your Oracle OCI 9 plug-in you should select the Oracle OCI option when prompted to choose plug-ins to install.

When using Oracle 9i or 10g in parallel jobs on HPUX 11.00 or 11.11, you must add the additional paths - see "[Using Oracle 9i or 10g in Parallel Jobs on HP-UX](#)" on page A-15.

Creating MetaStage Locator Tables

"Capturing Process Meta Data" in *DataStage Administrator Guide* provides details of how to set up locator tables. These are used during job runs to create fully qualified locator strings which are subsequently passed to MetaStage to identify particular table definitions.

You can now view and edit the locator tables through the **Locator** tab of the **Table Definition** dialog box.

To ensure that imported and captured table definitions always match, create locator tables in the source and target databases used by the DataStage job. The locator table must include the name of the computer, the software product, and the data store.

When you import table definitions from these databases into DataStage, a fully qualified locator string is created for them based on

the information in the locator table. This locator information remains with the table definitions when they are imported into MetaStage or captured by the Process MetaBroker.

Use the following procedure when working with locator tables:

- 1** Create locator tables in the source and target databases, as described in "*Capturing Process Meta Data*" in *DataStage Administrator Guide*.
 - 2** Import the table definitions for your job (not the table definition for the locator table) into the DataStage Manager, using one of the following:
 - ODBC 3.0 MetaBroker:
In DataStage Manager, choose Import->Via MetaBrokers..., and then select the ODBC 3.0 MetaBroker.
 - Plug-ins:
In DataStage Manager, choose Import->Table Definitions->Plug-in Metadata Definitions..., and then select the plug-in appropriate to your database.
 - ODBC Built-in:
In DataStage Manager, choose Import->Table Definitions->ODBC Table Definitions..., and then enter the appropriate information in the **Import Meta Data (ODBC)** dialog box.
- Note** Enterprise Edition users who want to take advantage of the locator table can use either the ODBC 3.0 MetaBroker or the ODBC Built-in to import table definitions. This functionality is supported for Enterprise Edition users only on Oracle, DB2 and Teradata databases.
- 3** In DataStage Designer, load the table definitions into the DataStage job.
Note Note: If you save the table definitions, the locator information is not available in the saved version.
 - 4** Import the job and table definitions into MetaStage using the DataStage MetaBroker.
 - 5** Run the job to capture the process meta data. MetaStage matches the captured table definition to the table definition imported from DataStage.

Capturing Process Meta Data - Locator Table setup

"*Capturing Process Meta Data*" in *DataStage Administrator Guide* provides details of how to set up locator tables. Owing to differences in the way databases handle the case of text in column names, the following may apply:

Databases with SQL_IDENTIFIER_CASE values of SQL.IC.UPPER or SQL.IC.LOWER.

The MetaStage_Loc_Info table requires the columns 'Computer', 'SoftwareProduct' and 'DataStore' be created in both quoted and unquoted variants. The information entered for both variants should be identical.

For example, for Oracle, the DDL will be:

```
CREATE TABLE MetaStage_Loc_Info("Computer" VARCHAR(64),  
    "SoftwareProduct" VARCHAR(64),  
    "DataStore" VARCHAR(64),  
    Computer VARCHAR(64),  
    SoftwareProduct VARCHAR(64),  
    DataStore VARCHAR(64))
```

Databases with other values for SQL_IDENTIFIER_CASE will only need the unquoted columns.

Windows Install Issues (Client and Server)

This section describes install issues that are particular to Windows platforms. Both server and client issues are included.

RTI Server must be stopped manually before running install

This section only applies if you have installed RTI on the Windows server you are upgrading and the server you are upgrading is earlier than Release 7.0.1.

Before installing this version on a server that has RTI installed, the following steps should be carried out:

- Stop all RTI servers which use this DS server (see RTI documentation to know how to stop an RTI server).
- Then, on the DS Server host machine, through the Services Control Panel:

- a** If one of the RTI Servers is running on the same Windows system as the DS server, set the Startup Type for the RTI Server service to "Disabled"
- b** Reboot the system

After the install is complete, the following steps should be carried out:

- If you had to go through step a above, using the Services Control Panel, set the Startup Type for the RTI Server service to "Automatic".
- Start all the previously stopped RTI Servers (see RTI documentation) .

Windows 2000: Access required to the temp directory

DataStage requires that users have write permissions on the Windows temporary directory (e.g. c:\winnt\temp). On Windows 2000, non-Admin users do not have this permission set by default. Therefore, for non-Admin users, the c:\winnt\temp directory needs to be set up to have full access permissions for all users.

Windows 2000: Login Policy must be set

By default Windows 2000 does not let any non-Admin users connect to the machine. To overcome this, the login policy needs to be set appropriately. This can be done using the Start->Programs->AdministrativeTools->LocalSecurityPolicy tool.

Find the **log on locally** policy in the SecuritySettings->LocalPolicies->UserRightsAssignment folder and add the required users.

Failure to Register .ocx on Some Systems

After finishing the Client install, you may be asked if you would like to reboot your machine. If you answer Yes, on some systems, the following error is returned:

```
RegSvr32 DllRegisterServer in  
<drive>:\ProgramFiles\Ascential\DataStage\SQLBld0cx.ocx failed Return  
code was: 0x80040201.
```

This appears to be caused by a mismatch of system dll's on the system and is documented as a known issue on the Microsoft support website, as follows :

PRB: Mismatched 01e32.dll and Comcat.dll

<http://support.microsoft.com/support/kb/articles/Q201/3/64.ASP>

This states that "Comcat.dll version 5.0 works with Ole32.dll version 4.71 and later. Comcat.dll version 4.71 works with Ole32.dll version 4.0."

The workaround for this problem is to rename the "Comcat.dll" file and then install the "Msvbvm50.exe" package from Microsoft. This can be obtained via the following link:

<http://support.microsoft.com/support/kb/articles/Q180/0/71.ASP>

Once this has been done, the DataStage clients should then be able to be re-installed successfully.

Windows 2000 Advanced Server: Importing Active/X Routines

Windows 2000 Advanced Server has increased security over other Windows releases. This causes a problem when trying to import External Function Definitions from the DS Manager, when logged in as a non-Admin user.

The error reported is : "Error accessing the OLE registry". In order to prevent this error, the Registry must be updated to grant access to the appropriate user(s), as follows :

- Grant the (user or user's group) "Full Access" to the HKEY_CLASSES_ROOT registry hive. Use regedt32.exe in the WinNT\SYSTEM32\ directory to do this - The option can be found by selecting the top of the HKEY_CLASSES_ROOT tree, and then selecting the Security option from the menu.

External Active-X Routine arguments

The restriction to routine arguments described in "[Transform Routine Arguments](#)" on [page A-30](#) also applies to External Active-X Routine arguments. In order to protect against the direct modification of input column variables within these routines, it is suggested that arguments to external Active-X routines are defined as "call by value" rather than "call by reference". This will protect against inadvertent modification of a passed-in input column variable within the external routine. External routine arguments can be defined as "call by value" within the definition of the library containing the external routines (this is outside of DataStage control). Re-importing the routines after such a modification will ensure that the DataStage routine wrappers are correctly defined as "call by value".

Windows 2000 Adv Server: Reporting Assistant update failure

By default, Windows 2000 Adv Server restricts access to the c:\Program Files directory to Admin users. This causes a problem with the Reporting Assistant and Documentation tool when trying to Update the reporting database, when the DS clients are installed in the default location.

The workaround is to either install the clients into a directory which is configured to provide full access to all appropriate users or to modify the permissions on the c:\Program Files directory as appropriate.

Note that this also affects Export from the Manager & Designer, as the default directory for export files is the client install directory.

Documentation Tool Install Issues

The following issues concerning the installation of the Documentation Tool (MS Access components) should be noted:

- If you install the Documentation Tool (specifically the "Reporting Tool" Access application) to an ENGLISH system without MS Access, then the Access runtime components are installed to a directory \Program Files\Microsoft Office\Office. On reinstalls, the Access runtime components are not installed as the machine thinks it has Access runtime already installed (i.e. the one you just deleted). If this is the case, you can get round this by deleting the following registry key:

```
HKEY_CLASSES_ROOT\Access.Application.
```

Then the DataStage Client Setup program will reinstall the Access runtime too, updating the relevant registry entries. This situation can also occur if you reinstall the tool into a different location (leaving the original install on the machine), then remove the original install.

- If you are upgrading a DataStage 3.6 client system which has the Documentation Tool installed, the following Access keys should be removed from the Registry before installing the new clients:

```
HKEY_CLASSES_ROOT\Access.Application
```

```
HKEY_CLASSES_ROOT\Access.Application.8
```

Failure to do this will mean that DataStage will report the following error when trying to update the Documentation Tool database from the Reporting Assistant:

Documentation Tool couldn't find file 'c:\Program Files\Ascential\DataStage\Documentation Tool\system.MDW. This file is required for startup.

Use of Virus Scanners on the DataStage Server

Owing to the intrusive nature of virus scanning programs when configured to scan all files on a system, it is suggested that this feature be disabled or tuned, to avoid possible performance impact of the DataStage server. Tuning suggestions include only scanning program files (it is possible in most cases to specify additional common file extensions) and configuring the virus scanner to avoid DataStage projects.

UNIX Install Issues

This section describes install issues that are particular to Windows platforms.

Administrative User

Releases of DataStage prior to 5.2 had to be administered by root. Release 5.1 onwards could be administered by the special user *dsadm*. From Release 7.0 onwards, you can specify a different administrative user. When upgrading from releases prior to 7.0, you will be able to administer DataStage as root or as *dsadm* (set up the *dsadm* user as described in ["Setting Up Users and Group"](#) on [page 3-6](#)). When upgrading from Release 7.0, you will be able to administer DataStage as root, *dsadm*, or any administrative user that you set up at the initial Release 7.0 install.

The dsenv File

dsenv is a centralized file for storing environmental variables that was introduced at Release 4.0. The file is a series of bourne shell arguments which are referenced during server engine start up, and can be referenced by interactive users and by other programs or scripts.

To emulate the environment of the DataStage server engine (in a bourne shell) execute the following command from the *DSHOME* directory.

```
. ./dsenv
```

Any environment variable that is required by the DataStage server engine should be placed in the *dsenv* file. The DataStage server engine should then be stopped and started. On new installations, the *dsenv* file contains default values for necessary variables.

Versions of DataStage on Unix prior to 4.0 set environment variables needed by the server engine in the *uv.rc* start/stop script. The *uv.rc* script now references the *dsenv* file found in *DSHOME/dsenv*. Upgrades from previous versions of DataStage will transfer the old information into the new *dsenv* file. In some cases, you may need to manually modify or correct the transferred information.

Reverting to a Previous Version of the DataDirect ODBC Drivers

It is important to note that it is not possible to use multiple ODBC driver sets concurrently.

If you are upgrading from a DataStage release earlier than 4.2 then the preserved driver files will be in a directory *branded_odbc* under the *UVHome* directory. Later releases will see a directory called *branded_odbcxxx* under the install directory, where *xxx* is the previous release number.

The configuration files *dsenv* and *.odbc.ini* are preserved with a date time suffix.

To revert to a previous driver set, proceed as follows:

- 1 Identify the path of the previous release of branded drivers and make a note of it.
- 2 Preserve the *dsenv* and *.odbc.ini* files by renaming them with a suitable suffix (e.g. *.ship*).
- 3 Reinstate the versions saved with the date time suffix.
- 4 Manually merge the *dsenv* file with that saved at step 2.
- 5 Ensure that all paths in the *.odbc.ini* file are correct. The drivers will be in the directory identified in step 1 above.

Ensure that the the path noted in step 1 above is added to the *LD_LIBRARY* path setting in the *dsenv* file

Change to Operation of dsjob on UNIX

From release 7.0 onwards, *dsjob* on UNIX now outputs the return code to the standard error stream in all circumstances.

On UNIX the return code will be 255 if the real return code was negative or greater than 254. If the return code is 255, then to access

the 'real' return code users must capture the stderr stream and process the text output.

Solaris Install Issues

Multi-Processor Systems

Before installing or upgrading DataStage on a multi-processor Solaris system 2.7 or 2.8 system, you should install one of the following Solaris patches:

- 106980-19 (Solaris 2.7)
- 108827-26 (Solaris 2.8)

Installing Enterprise Edition

If you are upgrading to include Enterprise Edition, you should install one of the following patches before upgrade:

- 106327/12 (Solaris 2.7)
- 108434/12 (Solaris 2.8)

Solaris 2.9

Before upgrading to this release on a Solaris 2.9 system, a patched version of cpio is required. This is available in the following T-PATCH:

- T113280-03

AIX Install Issues

You require the IBM Visual Age C++ Compiler Version 6 or IBM XL C/C++ v7 runtime libraries for this release of DataStage. These can be downloaded from the following site and should be installed before you upgrade:

- <http://www-1.ibm.com/support/docview.wss?rs=0&uid=swg24001467>

If you are upgrading to include Enterprise Edition, or are upgrading an earlier Enterprise Edition version, you require the full Version 6 compiler, not just the run time library.

If you are deploying parallel jobs to other systems, you should ensure that they have the runtime library installed.

HP-UX Install Issues

From release 7.0 onwards, DataStage Enterprise Edition requires the aCC 3.50 compiler with the patch PHSS_29483 available from <http://itrc.hp.com>. Installing this patch changes the version of the compiler from 3.5 to 3.52.

Compaq Tru 64 Install Issues

Mounting the CDROM

After mounting the CDROM, if the file names do not appear in uppercase followed by a ;1, you should re-mount the CDROM device using the -o ISO9660 option. For example:

```
mount -r -t cdfs -o ISO9660 /dev/disk/cdrom0c /cdrom
```

Compaq C++ Run-Time Library

If you are upgrading from releases earlier than 5.1 on a Compaq Tru64 system you **MUST** install the Compaq C++ Run-Time Library first. The library is provided as part of a redistribution kit, details of which can be found on the Compaq web site at the following location:

```
ftp://ftp.compaq.com/pub/products/C-CXX/Tru64/cxx/cxxredist.htm
```

The redistribution kit is supplied in the form of a tar file and can be downloaded from the following address:

```
ftp://ftp.compaq.com/pub/products/C-CXX/tru64/cxx/
```

It is recommended that you always download the latest redistribution kit.

Note that if you already have a C++ Compiler installed, or if you install one as part of DataStage Enterprise Edition configuration, you must ensure that any Runtime Libraries downloaded are fully compatible with the version of the compiler installed. Generally, if you have the correct compiler installed, you will not need to update the Runtime Libraries.

Problems and Restrictions - Release 7.0, 7.0.1. and 7.1 Onwards

These problems and restrictions still apply if you are upgrading from DataStage Release 7.0, 7.0.1, or 7.1:

Time Based Job Monitoring - Intermittent Problems

Intermittent problems have been observed while running jobs on the Parallel canvas when time based job monitoring is enabled (the default). Time based job monitoring can be disabled in favor of size based job monitoring. This is done by unsetting the `APT_MONITOR_TIME` environment variable and setting the `APT_MONITOR_SIZE` variable to a suitable number, e.g. 1000000. This will cause the job to update row count information every 1000000 rows. The environment variables can be set in the Project Properties (in the Administrator) – this will affect all jobs. Alternatively, they be set for an individual job using the Job Properties screen in the Designer.

Performance Improvement for Enterprise Edition on Tru 64 Systems

In order to improve performance of parallel jobs, we recommend that you disable their use of shared memory by setting the `APT_PM_NO_SHARED_MEMORY` environment variable to 1. See ["Setting Environment Variables"](#) in *DataStage Administrator Guide* for help with this.

Oracle 7 OCI Plug-in Not Supported

The Oracle 7 OCI plug-in is no longer supported. This is because Oracle no longer support Oracle 7.

Oracle OCI8i Plug-in No Longer Supports Oracle 8.1.6 on AIX

Oracle 8.1.6 is not supported by the Oracle OCI8i plug-in from release 7.0.1 onwards (this is due to a conflict between the Oracle client library and the Java runtime library used by DataStage. In this situation, we recommend that you upgrade to the Oracle 8.1.7 client.

Using Oracle 9i or 10g in Parallel Jobs on HP-UX

When using Oracle 9i or 10g with DataStage Enterprise Edition on HPUX 11.00 or 11.11, you must add the following paths to your `SHLIB_PATH` before `ORACLE_HOME`:

```
$DSHOME/java/jre/lib/PA_RISC2.0  
$DSHOME/java/jre/lib/PA_RISC2.0/hotspot
```

LD_LIBRARY_PATH library path order for Oracle 8i BulkLoader Plug-in on TRU64

On the Compaq TRU64 platform, the order of the library path entries in LD_LIBRARY_PATH for the Oracle 8i BulkLoader plug-in has to be set as:

```
LD_LIBRARY_PATH=`cat /.dshome`/lib:odbc_driver_path:$ORACLE_HOME/  
lib:$LD_LIBRARY_PATH.
```

This is so that the Oracle client library generated built by the *genclntsh* script (mentioned above) is loaded first to resolve the necessary Oracle symbols.

AxcelPack

AxcelPack functionality has now been incorporated into the main DataStage release.

Problems and Restrictions – Release 6.0 Onwards

These problems and restrictions still apply if you are upgrading from DataStage release 6.0.

UniData Stage - Fails to write to a UniData Table

A job which contains two UniData stages connected directly together may fail with the following error:

```
jobname.linkname: ds_udtGetNextPacket Error  
calling DSHELPER subroutine - UniData Client error: call to UniRun  
returned 429  
- Argument at position 1 has invalid handle.
```

This is caused by UDT 6 not supporting multiple connections through ObjectCall from a single process. The workaround is to use the new UniData 6 stage.

DataStage Client/Server RPC Inactivity timeout

The Administrator client allows the user to change the Inactivity timeout value in the dsrpcservices file. Note that it is necessary to stop and re-start the services on the DataStage server before the change comes into effect.

If a job takes longer than the timeout period to complete then an error is reported, for example:

```
srvtrxCl,0: Unable to wait for job to finish - 81002.
```

This is logged as the job is finishing.

The default timeout is 1 hour, which means a timeout can arise with relatively short lived jobs. In order to overcome this eventuality, set the timeout value to a higher value, as described in the "[Setting the Server Timeout](#)" in *DataStage Administrator Guide*.

Setting SHMMNI on HP-UX Systems (UNIX)

This guide gives guidelines for setting the Kernel parameter SHMMNI, recommending a value of 2000. Note that HP do not recommend setting this parameter higher than 1024 in normal circumstances.

Problems and Restrictions – Release 5.2.2 Onwards

These problems and restrictions still apply if you are upgrading from DataStage release 5.2.2.

Defining Connections to Remote UniVerse databases

When defining a connection to a remote UniVerse database, a new `uvodbc.config` entry is added. It is important that the last line (`host = name`) is terminated with either a line terminator or space; otherwise the hostname does not get picked up correctly by the DataStage Engine. In this instance, error 81011 (bad hostname or host not responding) is reported.

Change to Input Stage Delimited Sequential Files

A change has been made to the handling of input stage delimited sequential files. The change in behavior is that NULL values will no longer be quoted, irrespective of the replacement token.

The old behavior was that, when a replacement string was specified, this was output unquoted. When no replacement string was specified, a pair of quotes was output.

The new behavior is that, when a replacement string is specified, this is output unquoted. When no replacement string is specified, nothing is output.

To revert to the previous behavior specify a replacement string of 4 quote characters, i.e., """".

Problems and Restrictions – Release 4.2.2 Onwards

These problems and restrictions still apply if you are upgrading from DataStage release 4.2.2.

Hashed File Stage Modifications

The Hashed File stage no longer caches writes for large files (modulus > 256) by default, as it did in previous releases. This is now controllable via the Hashed File stage editor GUI, on the **Inputs** page **General** tab.

In previous releases the size of the write cache was up to two times the modulus of the output file. However, in this release the size of the cache can be configured in the Administrator client.

Recommended not to use LAN Manager connectivity for import

It is possible that importing DataStage components from a large .dsx file when logged in using LAN Manager connectivity will crash. This is caused by a bug concerning the maximum size of data packets which can be transferred between client and server. The workaround is to use TCP/IP connectivity at the DataStage login dialog.

Enhanced Transaction Grouping Support

Release 4.2 onwards included changes which made the transactional properties of the UniVerse and ODBC stages more consistent. The transaction properties (Isolation level, Transaction size and Array size) have moved from the **Inputs** page **General** tab to the **Inputs** page **Transaction Handling** tab.

- 1** All links drawn from one transformer stage to one ODBC stage have always shared a connection/transaction context, but the point at which commits were performed was not always consistent. This set of input links is now known as a transaction group, either a single-link or a multi-link transaction group.
- 2** The single link case will be treated in much the same manner as it was at release 3.6 and earlier. The link between transaction size and array size which was introduced at release 4.0 has been relaxed. If these values differ than the behavior will be as per release 3.6. If the two values are the same then the release 4.0 functionality will be used. In the event of a write failure this functionality backed out the failed write and then proceeded to write a row at a time.
- 3** In the case of multi-link transactions they will either be available in the old release 3.6 functionality, complete with array binding and transaction size, or in the new grouped functionality. A multi-link transaction group does not support array binding; the array size property is not shown but its effective value is always 1. The transaction size of a multi-link transaction group is always 1. Existing jobs will be imported with transaction grouping disabled (i.e. using the existing pre-release 4.0 functionality) with the exception of jobs imported from 4.0.3 which is compatible with the functionality provided in this release and hence jobs from this release will maintain their current transaction grouping behavior.
- 4** Multi-link transactions are now supported within the UniVerse stage, but only for remote UniVerse connections.

DataStage Signal Handling Issue on Solaris (UNIX)

An issue exists with DataStage on fast multi-processor Sun systems that can cause a job to crash at runtime with a status of 96. This issue appears to occur with greater frequency as the number of DataStage jobs that are run simultaneously increases, and is not confined to a particular job design.

At the time of failure:

- 1** A core file is produced in the DataStage project directory.
- 2** A file that begins with "capture" followed by the actual pid of the DataStage process and a two character field is created and left in the directory that is pointed to by the UVTEMP parameter, which is defined in the *uvconfig* file. The default setting for the UVTEMP configurable parameter is */tmp*. An example of a capture file name is:
`capture14911aa.`

The failure is due to a signal handling issue between the Unix parent and child processes that occurs when the child process exits. This

issue has can be resolved on most Sun systems by removing the NOTIFY ON statement from the LOGIN paragraph that is located in the VOC file of each DataStage project. Since the standard LOGIN paragraph does not contain any other relevant commands it is safe to remove it by issuing a DELETEVOC LOGIN command from the DataStage engine > command prompt while in the project directory. The same task can also be accomplished by utilizing the **DataStage Administrator>Projects>Command option**.

If the above procedures do not resolve the issue, a fix can be downloaded from Sun that ensures that the signal handling code does not try to execute a signal handler if the specified address is 00.

A patch for Solaris 2.7 should be available. The fix is included in Solaris 2.8.

To determine if you are affected by the issue:

- Ensure that a core file and a capture file exist and that their creation times are identical.
- Use adb or dbx to see if the core was due to a SIGSEGV: Segmentation Fault, caused by executing code at address 0.

Oracle OCI 8 Plug-in Support for Operating System Authentication

The OCI 8 plug-in now provides support for Operating System authentication. A new checkbox labelled **Use OS level authentication** has been added to the **Input** page **General** tab. By default the box is unchecked and the stage uses data entered in the **User ID** and **Password** fields. If the box is checked the **User ID** and **Password** fields are ignored; authentication is performed using the **User ID** and **Password** of the user executing the DataStage job. Note that it is still possible to enter data into the **User ID** and **Password** fields even if the box is checked, but the data is ignored.

Oracle OCI 8 Plug-in Pre 4.2 User-defined SQL behavior

The OCI 8 plugin has a new property: **Pre 4.2 User-defined SQL behavior**. It is presented as a checkbox in the Custom GUI on the **Input** page **General** tab and **Output** page **General** tab. The checkbox is not visible by default, it depends on the Update action (for Input) and the Query type (for Output when the stage is used as a reference lookup).

- **Input page General tab.** On the **Input** page **General** tab the new property can only be accessed by setting the **Update** action to the following values one after the other:

- Set the **Update** action to **Delete existing rows only**
- Set the **Update** action to **User-defined SQL**

This will display the **Pre 4.2 User-defined SQL behavior** checkbox which will be checked by default.

- Output page General tab. The Pre 4.2 User-defined SQL behavior is only available on this page when the plug-in is used as a reference lookup stage. To display the property, set the Query type to User-defined SQL query. The property will be checked by default.

When the **Pre 4.2 User-defined SQL behavior** property is checked the parameter binding for a SELECT or DELETE SQL statement will behave as it did prior to DataStage 4.2, which means that the parameter binding always starts at :1 for the WHERE clause. When the checkbox is not checked, the parameter binding is key column position dependent, e.g., SELECT ENAME, EMPNO FROM gtar34508 WHERE EMPNO = :2, or DELETE FROM gtar34508 WHERE EMPNO = :2.

The DELETE SQL statement has to be key column position dependent when it uses the WHERE clause and is combined with another statement: DELETE FROM gtar34508 WHERE EMPNO = :2; INSERT INTO gtar34508 (ENAME, EMPNO) VALUES (:1, :2)

This is because only one parameter binding happens for these two SQL statements. Only these two SQL statements are supported in this combination for an input link. One SQL statement is supported for a reference/output link.

Problems and Restrictions – Release 4.2 Onwards

These problems and restrictions still apply if you are upgrading from DataStage release 4.2.

Installation of Developer Studio onto DataStage clients (Windows)

When installing Visual C++ 6 or Visual Basic 6 after installing the DataStage clients, make sure that you also use the Microsoft Visual Studio Service Pack 5

Defining Sequential File Meta Data

When creating a sequential file definition in the Manager, you should click OK once you have entered column definitions in the **Columns** tab. If you revisit the **Format** tab before clicking OK, you will lose the column definitions.

Red Brick ODBC Driver Message Suppression (Windows)

DataStage 4.2 does not support the additional informational messages (SUCCESS_WITH_INFO) which are available from the Red Brick ODBC Driver. The driver must be configured to suppress these messages as follows:

On Windows you will need to modify the Red Brick ini files used to pass optional configuration parameters to the driver.

Two files exist if both 32 bit and 16 bit drivers are installed:

- rbodbc32.ini - 32 bit driver
- rbodbc16.ini - 16 bit driver.

The files are located in the windows install directory (e.g., c:/winnt directory).

The ini files should be modified to insert the following line into the file:

```
DISABLE_INFO=YES
```

For example, the rbodbc32.ini file will look similar to the following after applying the change :

```
[SQL]
INSTALL_DIR=C:\RedBrick\Client32\
LOCALE=English_UnitedStates.US-ASCII@Binary
MESSAGE_DIR=C:\RedBrick\Client32\messages
ANSI_DATETIME_DISPLAY_FORMAT=OFF
DISABLE_INFO=YES
```

Packager Installer log files

In some circumstances, the Package Installer does not report errors and you should always check the log files created during the installation to ensure that the install was successful.

Modified Sequential File stage handling of Numeric & Decimal

The Sequential File stage now uses scale and precision when writing out values of SQL types Numeric and Decimal.

Jobs at 3.6 with no scale set (= scale zero) used to produce 4 decimal places (a UniVerse default). The result now is that no decimal places are produced; the number will be rounded to the nearest integer.

DataStage Enterprise MVS Specific Issues

This section lists issues specific to Enterprise MVS Edition and mainframe jobs.

Release 7.x Enhancements

Nullable Columns in Delimited Flat File Target Stages

From release 7.5 onwards, you may see different results for nullable columns when targeting a Delimited Flat File stage. In previous releases, if a column coming into a Delimited Flat File stage had been defined as nullable and the null indicator was turned on, the column was written with a value of character 0 (zero) or blank, depending on the column data type. From release 7.5 onwards, the column will not have a value; instead you will see two column delimiter characters. If the column is defined as not nullable or as nullable but the null indication is turned off, then the column will be written with character zero or blanks as was done before.

SUBSTRING Enhancements

From release 7.0 onwards, two SUBSTRING functions now exist: SUBSTRING and SUBSTRING_MB. If you were using in bounds values for the StartPosition and StringLength, the new SUBSTRING function will give the same results as the old function. However, if you used out-of-bounds values for these arguments (e.g., zero or negative values or values too large), you may get different results with the new SUBSTRING function.

Release 6.0 Enhancements

The following enhancements were made to Enterprise MVS Edition (DS XE/390 as it was then called) at Release 6.0.

Delimited Flat File Source Stage

The Delimited Flat File stage was extended to support reading data from delimited flat files. As a source, the stage can have multiple output links and users can specify constraints to filter output data.

New Teradata Stages

Three new stage types were added to read data from and write data to Teradata databases: Teradata Relational, Teradata Export, and Teradata Load.

Teradata Relational stages can be used as a source or a target, while Teradata Export stages are source stages only, and Teradata Load stages are target stages only. The Teradata Load stage supports the FastLoad, MultiLoad, and TPump loading utilities.

Pre-sorting of Source Data

Source data being read in Complex Flat File and Fixed-Width Flat File stages can now be pre-sorted to simplify processing in later, active stages where data transformations and aggregations are performed.

Support for Variable-Block Files

Complex Flat File and Multi-Format Flat File stages were modified to read data from source files which have either variable or fixed record lengths, with or without OCCURS or OCCURS DEPENDING ON clauses.

Multiple Record Type Merge in Multi-Format Flat File Stages

The Multi-Format Flat File stage was enhanced to allow multiple record types to be output on a single output link.

End-of-data Indicators in Source Stages

Users now can add an end-of-data row after the last row is processed on each output link from all mainframe job source stage types.

Target File Deletion Option

A new write option, **Delete and recreate existing file**, was added to Fixed-Width Flat File, DB2 Load Ready Flat File, and Delimited Flat File target stages. This allows flat files to be deleted (if already cataloged) prior to actual use when the same job is run multiple times.

Tape Expiration Date

Fixed-Width Flat File, DB2 Load Ready Flat File, and Delimited Flat File target stages now provide a file deletion option for a new data set. Either an expiration date or a retention period may be specified.

Conditional Lookups

The Lookup stage was enhanced to support conditional lookups, where a pre-lookup condition must be met before a lookup is performed. This can improve job performance by skipping the execution of a lookup if the data is not needed or is already available.

Connect:Direct File Exchange

The FTP stage now supports Connect:Direct as a file exchange method in addition to FTP.

String Function Enhancements

New LPAD and RPAD string functions were added to this release, allowing for left and right padding to occur in expressions. In addition, the SUBSTRING function was enhanced to support either character-oriented or byte-oriented substring extraction.

HIGH_VALUES and LOW_VALUES Support

Two new constants named HIGH_VALUES and LOW_VALUES were added to the Transformer stage Expression Editor. They can be used alone in a derivation, as the initial value of a stage variable, or as the argument of a CAST function.

Program and Data Flow Tracing During Code Generation

To provide greater debugging support, a new option was added to the **Code Generation** dialog box, allowing the program and/or data flow to be printed during program execution.

Design-Time Semantic Checking

The mainframe Expression Editor now checks for semantic errors as well as syntax errors. An option called **Perform expression semantic checking** has been added to both job and project properties.

Extended Decimal Support

To support DECIMAL data with precision great than 18, there is a new option called **Support extended decimal** in project properties. Users can specify the maximum decimal size that is allowed by their COBOL compiler.

Column Value Propagation

Column values can now be propagated via the right mouse button on the Columns grid of any mainframe stage editor (except those that are read-only). This makes it easy to apply changes to groups of columns in mainframe jobs.

Release 6.0 Problems and Restrictions

The following issues apply to DataStage Enterprise MVS Edition from Release 6.0 onwards.

New Enterprise MVS Edition Runtime Library (RTL)

With Version 6.0.1 of Enterprise MVS Edition (DataStage XE/390), the interface to the mainframe RTL routines (COPYLIB and LINKLIB) was changed. The new interface is not backward compatible with previous versions of XE/390 (5.2.1 or earlier). Therefore, you must use the latest version of the two mainframe RTL libraries (COPYLIB and LINKLIB) that are shipped with 6.0.1.

Release 5.2.2 Enhancements

The following enhancements were made to Enterprise MVS Edition (DS XE/390 as it was then called) at Release 5.2.2.

Time to Decimal Mapping

Earlier versions of the *Mainframe Job Developer's Guide* stated in Appendix B that Time to Decimal mappings are not allowed. This is incorrect. The mapping is performed using COBOL statements to convert the Time to a Decimal that represents the number of seconds since midnight. The precision of the Decimal must be at least 5.

Timestamp to Decimal Mapping

Earlier versions of the *Mainframe Job Developer's Guide* gives incorrect precision and scale values for the Timestamp to Decimal mapping described in Appendix B. The precision of the Decimal should be 18. If you want to include milliseconds, the scale of the Decimal should be no more than 6.

NChar and NVarChar Data Types

From Release 5.2.2 onwards, DataStage now provides native support for NChar and NVarChar data types. Existing jobs or table definitions

with GRAPHIC fields must be recaptured/redefined for the changes to take effect.

Job Log Messages Are Summarized

Messages in the job log are now summarized for updates and deletes to DB2. Instead of receiving row-by-row messages for each action performed, a single message reports the updates and deletes on a batch basis.

Release 5.2.2 Problems and Restrictions

The following issues apply to DataStage Enterprise MVS Edition from Release 5.2.2 onwards.

Using Job Parameters in Relational WHERE Clauses

The use of job parameters in Relational stage WHERE clauses differs depending on whether the stage is a source or a target. Job parameters are available on the Where tab of Relational target stages, but they do not appear on the Where tab of Relational source stages. To use a job parameter in the SELECT statement of a Relational source stage, you can add a Transformer stage after the Relational stage in your job design, and define a constraint that compares a source column to the job parameter.

The source column and job parameter must have the same data type and length, no implicit or explicit casting can occur, and there cannot be an ORDER BY clause in the SQL SELECT statement. If these requirements are met, DataStage pushes the constraint back to the SELECT statement and the filtering is performed by the database.

If the source column and job parameter are not the same data type and length, you can convert them using one of the following techniques.

The first technique is to create a computed column in the Relational stage that has the correct data type and length, and use one of the DB2 functions to convert the original column to the data type and length of the job parameter. You can then use this computed column in the Transformer stage constraint.

The second technique converts the job parameter to the data type and length of the Relational column. For example, suppose you have a job consisting of a Relational source stage, a Transformer stage, and a Delimited Flat File target stage. You wish to use a job parameter called INDATE that represents the starting date for a database search. Since job parameters cannot be of Date type, you define it as Char, 10. The

Relational source column it will be compared to is called COLDATE and is defined as Date, 10. To push the job parameter to the SQL SELECT statement, you CAST the job parameter as a Date in a Transformer stage constraint, using an expression similar to this:

```
Tlinkname.COLDATE >= CAST (JobParam.INDATE as DATE)
```

When code is generated, the EXEC SQL statement will include the job parameter in a manner similar to this:

```
EXEC SQL
  DECLARE RELATIONAL_01 CURSOR FOR
  SELECT RELATIONAL_0_DSL_0.COLDATE,
  RELATIONAL_0_DSL_0.COLKEY,
  RELATIONAL_0_DSL_0.COLINT,
  RELATIONAL_0_DSL_0.COLDEC
  FROM (SELECT JOBPAR1.COLKEY,
```

Release 5.2 Problems and Restrictions

The following issues apply to DataStage Enterprise MVS Edition from Release 5.2 onwards:

Changes to the Precision of Decimal, Integer, and SmallInt Data

From release 5.2 onwards, when importing DCLGen files, DataStage now captures Decimal data with precision up to 31, Integer data with precision up to 10, and SmallInt data with precision up to 5. This is for consistency with the data capture techniques used in the DataStage server edition and third-party tools.

However, when loading such data into mainframe jobs, any columns with data type Decimal, Integer, or SmallInt which have precision greater than 18, 9, and 4, respectively, are changed to the precision 18, 9, and 4 due to COBOL restrictions. You may need to change the data type of your source data or perform data type mappings within DataStage to work around this restriction.

Release 5.1 Problems and Restrictions

The following issues apply to DataStage Enterprise MVS Edition from Release 5.1 onwards:

Nullability of Complex Flat file and Multiformat Flat File Stage Columns

The stage columns in Complex Flat File and Multiformat Flat File stage are expected to be always not NULL.

Delimited Flat File stage

Column and string delimiters in the Delimited Flat File stage must be only printable characters in the range 0 - 255.

Date masks on columns in relational stage used as target

Before a job is imported from previous versions to this release with a process (active) stage followed by a relational stage and with the column data type as DATE in the process stage, perform the following steps in the dsx file:

- 1 Delete the DateMask for the column
- 2 Set the SqlType to 1
- 3 Then import

DataStage Enterprise Edition Specific Issues

Recompiling Parallel Jobs

If you are upgrading from a Pre-Release 7.0 release, you will need to recompile all your parallel jobs after the upgrade is complete. You can use the batch job compilation wizard for this, or a command line interface. See "[Compiling Server Jobs and Parallel Jobs](#)" in *DataStage Designer Guide*.

Regeneration of Filesets

If you are upgrading from a Pre-Release 7.0.1 release, you will need to regenerate lookups created with previous versions of DataStage. This is because optimizations have been made to the variable length string handling in the parallel engine, and these optimizations are not backwards compatible.

Change to Final delimiter behavior for Parallel jobs

This applies if you are upgrading from a release before 7.1. This affects the Final delimiter format property on the **Format** tab. If you have a subrecord or tagged column that is the last column of the record (at that level, i.e. it won't be the very last column, since its last sub-column will be), then the previous behavior was that, even if Final delimiter was set to "none", a Delimiter (defined in the property of that

name) was inserted at the end of the subrecord/tagged column. This behavior was incorrect. The new default behavior is for this not to happen unless subrecord or tagged column is also a vector, in which case the previous behavior applies.

The new default behavior can be reverted to the previous behavior by setting the `APT_PREVIOUS_FINAL_DELIM_COMPATIBLE` to 1 in the Administrator client. Note that this will affect all job runs. To set this environment variable for a specific job only, first define the environment variable in the Administrator client with a default of empty-string. This will ensure that the environment variable is defined for the project, but not set and will therefore not affect the normal behavior. A job can then be set to use the previous behavior by using a job parameter tied to this environment variable, setting the default value to 1.

SyncSort Option Removed from Parallel Sort Stage

The parallel Sort stage no longer supports the SyncSort sorting utility. You must use either the DataStage or UNIX sort utility instead.

DataStage Server Job Specific Issues

Transform Routine Arguments

Following the introduction of a number of performance enhancements in the DataStage server, it is important that user written Routines are not written in such a way as to directly modify their input arguments if those arguments are actually input column variables. For example, input column variables are passed into Routines as arguments when called from a column derivation in the Transformer.

If an input column variable is passed into a Routine as an argument and then directly modified within the Routine, there is likely to be corruption of that input column data on returning from the Routine.

Note that Routines which modify input arguments which are not input columns variables are not affected by this restriction.

In order to check that the Routines contained in a project do not break this restriction, the DataStage Server installation has been modified to run a Routine checker utility which will report on any Routine which appears to be modifying it's input arguments. The utility is run against each project on installation and the results are output to a file named `CheckRoutines.log` in the project directory on the Server.

Note that the utility will report on any Routine which modifies its input arguments and as noted above, it is only those Routines whose arguments are actually input column variables which need to be modified.

The following example demonstrates a Routine which breaks this restriction and how it can be modified:

The Routine checks Arg1 to see if it's NULL and if so, sets it to be an empty string. It then goes on to use that string in some way:

```
MyRoutine ( Arg1 )
{
  If IsNULL(Arg1)
    Arg1 = " "
    ...go on to use Arg1 in some string processing
    operation...
  RETURN (Ans)
}
```

When used from a job, an input column variable is passed in as Arg1 and directly modifying it in this way is illegal. The Routine can be changed to take a local copy of the input column variable so as not to directly modify it, as follows :

```
MyRoutine ( Arg1 )
{
  InputValue = Arg1
  If IsNULL(InputValue)
    InputValue = " "
    ...go on to use InputValue in some string processing
    operation...
  RETURN (Ans)
}
```

Note that there are two Routines in the ClickPack which (if installed) will be included in the output from the checker utility. These are intended to work in this way and do not need to be modified. These Routines are:

- Transform Routine CPParseRequest in category ClickPack\Parsing
- Transform Routine CPParseURL in category ClickPack\Parsing

New CRC32() Function Available

A new CRC32() function was introduced at release 5.1 to create 32-bit cyclic redundancy codes. The interface and behavior are similar to the existing CHECKSUM() function. The function can be accessed through the Transformer stage. You may wish to upgrade your jobs to take advantage of this.

Active to Active link Row Buffering - Optional Performance Enhancement Introduced at Release 5.1

The DataStage runtime has been modified to provide an optional enhancement to the way row data is passed between linked active stages (e.g. a Transformer to Transformer link) in server jobs. This enhancement involves buffering of row data between the active stages, which can significantly improve performance.

Row buffering is turned off by default owing to a restriction regarding the use of routines or transforms referencing routines that use COMMON blocks to store and retrieve values. For more details see "[Optimizing Performance in Server Jobs](#)" in *Server Job Developer's Guide*.

UniVerse and Hashed File Stages - Compatibility between Releases

For users of the UniVerse or Hashed file stages there are some important differences to understand regarding their compatibility with revisions of DataStage. These are shown in the following table:

Feature	Release 6.0.1	Release 5.2	Release 5.1	Releases before 5.1
dsrpc port number	31538	31438	31438	31438
UvNet supported?	No	No	No	Yes
Incoming UniVerse stage connections supported	No	No	No	Yes
Outgoing UniVerse stage connections supported	to releases before 5.1			
Can read UniVerse hashed file ¹	No	No	No	Yes
Can read DataStage hashed file ²	Yes	Yes	Yes	No

1 A UniVerse hashed file is one created by UniVerse or by releases of DataStage prior to 5.1.

2 A DataStage hashed file is one created by DataStage from release 5.1 onwards.

Hash File Write Errors

From DataStage Release 7.5 onwards, failures that occur when writing to files using the Hashed File stage are logged (in previous releases, write failures were ignored).

UniData 6 Stage

A new UniData 6 stage allows access to IBM UniData 6 databases (not on Linux or Tru64). See *Accessing UniVerse and UniData Databases from DataStage* for detail.

Index

Symbols

/etc/magic 6-11

A

adding

 new Plug-ins 2-12, 4-5

 new projects 4-5

adding new 3-22, 5-13

advanced log reader stage 3-1

AIX 3-7, 3-12, 3-16, 3-28, 3-31, 5-5, 5-7, 5-21

AIX-specific kernel parameters 3-8

B

BCPLoad plug-In 3-31, 5-22

C

C++ compiler 6-13

C++ compilers 3-13

CD-ROM contents

 UNIX platforms 3-5

checklist

 UNIX install 3-1

 UNIX upgrade 5-4

checks

 pre-upgrade 5-5

ClickPack 2-3, 3-5

clients

 installing 2-12, 3-23

 upgrading 4-6, 5-15

Compaq Tru64 3-6, 3-7, 3-12, 3-16, 3-28,

 3-31, 5-5, 5-6, 5-7, 5-21, 10-4, 10-5

 runtime library 3-11

configuring 3-30

 ODBC access 3-32, 5-23

 Plug-ins 3-30, 5-22

configuring parallel engine 6-1

copylib 7-2

coresident systems 2-14, A-1, A-2

D

DataStage documentation xi

DataStage server 9-4

DB2/UDB databases 6-14

deadlock daemon 10-5

disk space

 UNIX platforms 3-12

 Windows platforms 7-4

documentation conventions x

documentation tool 2-14

dsadm user 3-6, 5-5

dsenv file 5-20

dsrpcd daemon 10-2

E

Enterprise Edition 2-2, 3-4

Enterprise MVS Edition 2-2, 3-4, 7-1

environment variables 3-28, 5-20, 6-11, 8-6

F

file units

 running out of 10-1

G

GCI 3-12

H

hardware and software requirements

 UNIX platforms 3-12

 Windows systems 2-3

HP-UX 3-7, 3-12, 3-16, 3-28, 3-31, 5-5, 5-7,

 5-21, 5-22

HP-UX-specific kernel parameters 3-9

-
- I**
- impersonation mode install 3-1, 5-2
 - Informix XPS databases 6-20
 - install checklist
 - UNIX platforms 3-1
 - install program
 - UNIX platforms 3-16, 5-7
 - install script
 - UNIX 3-12
 - installation directory
 - UNIX platforms 3-7
 - installing
 - clients 2-11, 3-23
 - DataStage components on mainframe platform 7-1
 - UNIX platforms 3-1
 - Windows platforms 2-1
 - installing server multiple instances 5-3
 - IVP 7-4
- J**
- Java Pack 2-3, 3-4
- K**
- kernel parameters 3-7
 - AIX-specific 3-8
 - HP-UX-specific 3-9
 - Tru64-specific 3-9
- L**
- LINUX 3-7, 3-16
 - log files 3-23, 5-15
- M**
- mainframe components
 - installing 7-1
 - mainframe IVP 7-4
 - MVS IMS Source 2-3, 3-5
- N**
- new DataStage server 9-4
 - new features 7-1, 9-1
 - NIS 3-12
 - non-impersonation mode install 3-2
- O**
- object call install 10-5
 - ODBC
 - configuration 3-32, 5-6
 - non-wire protocol 3-32
 - wire protocol 3-32
 - ODBC access
 - configuring 3-32, 5-23
 - Oracle databases 6-17
 - Oracle OCI Plug-ins 10-3, 10-5
- P**
- package installer 3-5
 - paging space 6-8
 - PAM 3-29, 5-19
 - parallel extender
 - configuring 6-1
 - PEPs
 - adding new 3-22, 5-13
 - permissions and security 5-19
 - per-user resource limits (Compaq Tru64) 6-6
 - Plug-ins 2-12, 3-22, 3-30, 5-13
 - configuring 5-22
 - post-install checks 3-26
 - pre-install checks
 - UNIX platforms 3-6
 - Windows platforms 2-1
 - pre-upgrade checks 5-5
 - projects
 - adding new 3-22, 5-13
- R**
- reinstalling 2-12, 3-12, 3-22, 5-13
 - reporting facilities 2-14
 - requirements
 - UNIX platforms 3-12
 - Windows platforms 7-4
 - restrictions 10-4
 - routines 5-1, 7-1, 9-1, 10-1
 - rsh 6-1
 - RTI Agent 3-5
 - RTI Agent. 2-3
 - runtime library (RTL) 7-2
- S**
- SAS Integration pack 2-3, 3-5
 - security and permissions 5-19
 - segment size limits (HP-UX) 6-7
 - Server 2-2, 3-4
 - setup program
 - Windows platforms 2-6, 4-2
 - Solaris 3-7, 3-12, 3-16, 3-28, 3-31, 5-5, 5-7, 5-21
 - starting DataStage 3-28, 3-39
 - Stopping DataStage 3-28, 3-39

- stored procedures 10–4
- supported platforms
 - UNIX 5–5
- supported platforms, UNIX 3–12
- swap space 6–8
- Sybase 10–5

T

- Teradata databases 6–19
- thewall parameter 6–4
- transforms 5–1, 9–1, 10–1
- troubleshooting
 - UNIX installations 10–1
- Tru64 3–31
- Tru64-specific kernel parameters 3–9

U

- UniVerse
 - coresident systems A–1, A–2
 - existing installations 2–14
- Upgrading 4–1
- upgrading 4–1
 - clients 4–6, 5–15
- users 5–19
- users and groups 3–6

V

- verifying
 - coresident installs 5–18, A–3

W

- Windows platforms 4–1

X

- XML Pack 2–3, 3–4

