Ascential **DataStage**

Accessing UniVerse and UniData Databases from DataStage

Ascential Software

Part No. 74-0121
December 2004

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

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

© 2005 Ascential Software Corporation. All rights reserved. DataStage®, EasyLogic®, EasyPath®, Enterprise Data Quality Management®, Iterations®, Matchware®, Mercator®, MetaBroker®, Application Integration, Simplified®, Ascential™, Ascential AuditStage™, Ascential DataStage™, Ascential ProfileStage™, Ascential QualityStage™, Ascential Enterprise Integration Suite™, Ascential Real-time Integration Services™, Ascential MetaStage™, and Ascential RTI™ are trademarks of Ascential Software Corporation or its affiliates and may be registered in the United States or other jurisdictions.

The software delivered to Licensee may contain third-party software code. See *Legal Notices* (legalnotices.pdf) for more information.

How to Use this Guide

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

This manual describes how to connect to UniVerse and UniData databases from a DataStage job.

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

Organization of This Manual

This manual contains the following:

- Chapter 1 describes how to connect to UniVerse databases.
- Chapter 2 describes how to connect to UniData databases.
- Chapter 3 describes how both databases describe multivalue data and multilevel files.

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.
Italic	In syntax, italic indicates information that you supply. In text, italic also indicates UNIX commands and options, file names, and pathnames.

Convention	Usage
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.
`This line ➥ continues`	The continuation character is used in source code examples to indicate a line that is too long to fit on the page, but must be entered as a single line on screen.

The following are also used:

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

DataStage Documentation

DataStage documentation includes the following:

- ***DataStage Administrator Guide***: This guide describes DataStage setup, routine housekeeping, and administration.

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

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

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

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

Contents

How to Use this Guide

Organization of This Manual iii
Documentation Conventions iii
DataStage Documentation iv

Chapter 1
UniVerse File and Table Access

Local and Remote Data Sources 1-1
Data Sources Local to the DataStage Server 1-1
Data Sources Remote from the DataStage Server 1-1
Co-resident Data Sources 1-3

Chapter 2
UniData File and Table Access

Accessing UniData 5.2 2-1
ObjectCall on UNIX Systems 2-1
Configuration File 2-2
Handling Large Records 2-6
UniData File Meta Data 2-6
Connecting to UniData 6 2-7
UniData 6 Account 2-7
DataStage DSN for UniData 6 Server 2-8
UniData 6 Connection Parameters 2-9
UniData Table Meta Data 2-10

Chapter 3
Multivalued Data and Multilevel Files

Multivalued/Multi-Subvalued Data and Associations. 3-1
Multiple Data Files (Multilevel Files) 3-4

1

UniVerse File and Table Access

Local and Remote Data Sources

Data Sources Local to the DataStage Server

The Hashed File stage provides access to hashed files. These are identified by **File name** and either **Account name** or **Account path**. The file name uniquely identifies a VOC record of type F or type Q in the specified account.

The UniVerse stage provides access to tables in the local SQL catalog and hashed files in the account specified by **Account name**. Omitting **Account name** defaults to the current DataStage project.

Data Sources Remote from the DataStage Server

The Hashed File stage provides access to UniVerse files on a remote UniVerse server that has UV/Net installed. A remote account can be accessed by **Account name** after creating an entry in the UV.ACCOUNT file of the form:

```
ID: REMOTE
008 Remote ACCOUNT Pointer
011 mk9!/export/home/paton
```

The record ID is the account name, field 8 can optionally contain descriptive text, and field 11 contains the UniVerse host name and the pathname of the account on the server system separated by a ! character.

Using UV/Net to access remote UniVerse files causes some confusion in the normally accepted client/server model. This is because, in general, the DataStage server itself becomes the client when

accessing meta data over the UV/Net connection. The server is then the machine on which UniVerse is being accessed.

Connectivity is further complicated because the initial DataStage client/server link can be made over either LAN Manager or TCP/IP. The former is achieved by selecting the **Omit** check box on the Attach to Project dialog box. The latter connection occurs when a user name and password are provided..

Attach Mode	DataStage Server	UV/Net Server	Comment
User name and password	Windows	Windows	Import possible (see note 1)
		UNIX	Import possible
	UNIX	Windows	Import not possible (see note 2)
		UNIX	Import possible
Omit option selected	Windows	Windows	Import not possible (see note 2)
		UNIX	Import possible
	UNIX	Windows	No connection possible (expected)
		UNIX	No connection possible (expected)

Notes

1. In order for the import to succeed it is necessary to specify the host name in the format *hostname*:*LAN* in the UV.ACCOUNT record.
2. Connection over UV/Net is possible from the DataStage server prompt, but a remote ID must be set up. This can be done using the SET.REMOTE.ID command or by setting a value in the UVNETRID environment variable. Neither approach works for meta data import as the *uvapi_slave* process does not handle environment variables or LOGIN paragraphs.

The UniVerse stage provides access to UniVerse tables in the SQL catalog of a remote UniVerse server and UniVerse files in an account on that system specified by **Account name**. An account path can also be used. Before you can do this you must create an entry in the *uvodbc.config* file in the DataStage project, which identifies the remote UniVerse server, of the form:

```
<mk9>
DBMSTYPE = UNIVERSE
network = TCP/IP
service = uvserver
host = mk9
```

The first line is the **DSN** enclosed in <> characters and the last line specifies the name of the remote UniVerse server.

Co-resident Data Sources

Accessing tables and files on a UniVerse system that resides on the same machine as the DataStage server is the same as accessing remote UniVerse data sources (see above). Specify the local hostname as the host in the uvodbc.config file entry.

2

UniData File and Table Access

Accessing UniData 5.2

Before any UniData 5.2 database can be accessed by DataStage, the appropriate UniData command must be run on the UniData server system to configure the ObjectCall server and the databases that can be accessed via it. The appropriate commands are UniServer Admin on Windows or *us_admin* on UNIX. The ObjectCall server/service may also need to be licensed and started.

The ObjectCall server must be configured to listen on port 4343 on UNIX servers. On Windows the port number can be configured in the uniapi.ini file.

Note Prior to release 7.0.1, DataStage servers running on HP-UX used port 4344 to access UniData 5.2. If you need to continue to use port 4344, you can do this by setting the UDTINI environment variable in dsenv next to the other UDT variables. UDTINI has the format:

```
PROTOCOL, internetaddress, portnnumber
```

For example:

```
13,127.0.0.1,4344
```

ObjectCall on UNIX Systems

On UNIX systems the ObjectCall server requires that the environment variables UDTHOME and UDTBIN are set up. We recommend that you set these in your *.login* or *.profile* file. Assuming the default names from a basic installation, the variables are set as follows:

- From the Bourne or Korn shell:

```
UDTHOME=/usr/ud41;export UDTHOME
UDTBIN=/usr/ud41/bin;export UDTBIN
```

- From the C shell:

```
setenv UDTHOME /usr/ud41
setenv UDTBIN /usr/ud41/bin
```

You also need to define UDTHOME and UDTBIN in the UniVerse *uv.rc* script (*uv.rc* starts the *unirpcd* daemon). The location and exact name of this script depends on the UNIX platform. Consult the DataStage *Readme* file for your platform for details.

To edit the *uv.rc* script to add the environment variables:

1. Find the text "USER MODIFIABLE PATH".
2. Modify the library path following this text to include the paths for UDTHOME and UDTBIN.
3. Restart the *unirpcd* daemon by killing the *unirpcd* daemon and running the *uv.rc* script.

You can use the ObjectCall client software on a UNIX platform other than the one where you have installed UniData and ObjectCall, provided both platforms are running the same operating system. To move ObjectCall to another UNIX platform:

1. Create a directory for the ObjectCall libraries on the platform you want to use as the client. You can use any name for this directory.
2. Move the following files from the server on which UniData is installed to the directory you created in step 1:

   $UDTHOME/lib/libmsg.a
   $UDTHOME/lib/libapi.a
3. Create a directory on the client platform to contain the ObjectCall ping program. You can use any name for this directory. Set the environment variable UDTBIN to this directory.
4. Move the file *$UDTHOME/bin/objping* from the server on which UniData is installed to the directory created in step 3.
5. Move the file *$UDTHOME/sys/uniapi.msg* from the server on which UniData is installed to the *sys* directory on the client platform.

Configuration File

When the DataStage 3.5 or later server is installed, a copy of the UniData API configuration file is installed.

On Windows servers, it is called *UNIAPI.INI* and is installed in the WINNT directory.

On UNIX servers, it is called *.uniapi* and is installed in your home directory (as defined by the environment variable HOME). If ObjectCall cannot find *.uniapi* in your home directory, it looks in */usr/udrel/include* (for example, on a system with UniData 4.1, this would be */usr/ud41/include*).

The configuration file has two optional sections:

- **Variables section**. Defines the settings of the environment variables used by ObjectCall.
- **Servers section**. Lists connection information for all the servers that the ObjectCall client can access.

The variables section allows you to set the following environment variables:

- UNILOGLEVEL.ObjectCall has a logging capability that is useful for debugging purposes. You can set the level of logging by adjusting the values of UNILOGLEVEL. Possible levels are:
  - 0 no logging
  - 1 log events of level UNIfatal
  - 2 log events of level UNIuserWarn
  - 4 log events of level UNIwarn
  - 8 log events of level UNItraceEnter
  - 16 log events of level UNItraceSuccess
  - 32 log events of level UNItracelog

  You can add these levels together to log events at several levels. For example, setting UNILOGLEVEL to 3 (1+2) logs both UNIfatal and UNIuserWarn events.
- UNILOGDIR. Specifies the location of the log file. On Windows servers, this is called *UNIAPIN.LOG* (where *N* is the ordinal of the task using ObjectCall) and the location defaults to the DataStage directory. On UNIX servers, the log file is called *uniapi.cprocess.id* (where *process.id* identifies the process using ObjectCall). The location defaults to the location specified by $TMPDIR. If $TMPDIR is not set, it defaults to */tmp*.
- UNIRESPONSESIZE. Specifies the maximum amount of data returned to the client by the server for a particular request. The maximum value is 20000000 (20 MB) and the minimum is 1024. You can use this value to help tune your system. With a larger value, the client may wait longer for the server to process a request for a large amount of data, but will receive a larger portion of the total request. With a lower value, the client will not have to wait so long, but will have to repeat the wait until it receives all the data.

- UNIPACKETSIZE. Specifies the maximum number of bytes that are transferred across the network during one send or receive. Multiple packets may be transferred to satisfy a particular client request. The default value is 1024. The maximum value is 2048 and the minimum 512.
- UNISVRSTATLEVEL and UNICLTSTATLEVEL. Specify whether status objects should be created on the server and client respectively. Possible settings are:
  - 0 no status object generation
  - 1 generate status objects for events of level UNIfatal
  - 2 generate status objects for events of level UNIfatal and UNIuserWarn
- PROTOCOL (Windows systems only). Specifies the version of the internal client/server protocol. It is set to 12 by default. If, when you first attempt to connect to your UniData server, you get an error message similar to:

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

  then you must edit the *UNIAPI.INI* file and change the value of the PROTOCOL variable. In this case, change it from 11 to 12:

  ```
  PROTOCOL = 12
  ```

Note that you cannot connect to servers requiring different protocols simultaneously.

The servers section lists connection information for the servers that you want the client to access. Each server is defined by four fields:

- Logical name. Specifies an alphanumeric string that is a unique identifier for a server. This is the name that you use when opening a server using the UniOpen function. (This identifies a logical server. UniData on a UNIX platform can accommodate multiple logical servers.)
- Internet Protocol Address. Specifies the IP address either in dot format (such as 1.1.1.1) or by the name of the platform where UniData and the server you are going to access reside.
- Port number. Specifies the port number of the server you are going to access. ObjectCall has officially reserved port number 4344 and we encourage you to use this.

After setting up server details in the configuration file, you can check the connectivity of the server.

On Windows systems:

1. Double-click the **ObjectCall Ping** icon in the ObjectCall program group. The ObjectCall Ping window appears.
2. Enter your user name and password for the server containing the database you want to test connectivity to, then click **OK**. The Servers window appears displaying the available servers defined in the *UNAPI.INI* file.
3. Click the server you want to test connectivity to.
4. If the connection succeeds, a message box confirms this. Click **OK**. (The message box will also inform you if the connection fails.) The Databases window appears displaying all available databases on the server.
5. Click the database you want to open.
6. If the connection succeeds, a message box confirms this. Click **OK**. (The message box will also inform you if the connection fails.)

On UNIX systems:

1. Execute the ObjectCall ping program. This is called *objping* and is located in *$UDTBIN*.
2. Enter your login name for the server platform running UniData and UniServer.
3. Enter your password.
4. The ObjectCall ping program displays the servers you have listed in your *.uniapi* file. Enter the number of the server that you want to try the connection to.

   If a connection cannot be made, an error message is displayed, otherwise “Ping of SERVER succeeded” is displayed.
5. If the ping succeeds, the program displays the available databases on that server. Enter the number of the database you want to check the connection to.

   If a connection cannot be made, an error message is displayed; otherwise “Ping of DATABASE succeeded” is displayed.

If you cannot open a server or a database, check your *UNIAPI.INI* or *uniapi* file and the server administration in UniServer Admin or *us_admin*.

You do not have to name a server in the configuration file in order to access it. You can set the properties of a UNIclient object to specify the information needed to make a connection.

The following is an example *UNIAPI.INI* file:

```
[variables]
PROTOCOL = 12
UNILOGDIR = D:\Ardent\DataStage
;UNILOGLEVEL = 0
```

```
;UNICLTSTATLEVEL = 2
;UNIRESPONSESIZE = 4000000
;UNIPACKETSIZE = 1024

[servers]
;uniapi = PROTOCOL, 1.1.1.1, 4343
```

The following is an example *.uniapi* file:

```
[variables]
#UNILOGDIR = /tmp
#UNILOGLEVEL = 0
#UNISVRSTATLEVEL = 2
#UNICLTSTATLEVEL = 2
#UNIRESPONSESIZE = 4000000
#UNIPACKETSIZE = 1024

[servers]
uniapi = PROTOCOL, 1.1.1.1, 4343
```

Lines in the variables section can be commented out using the ; or the # character, causing that variable to take the default value. To activate the variable and set a different value, remove the ; or #.

Handling Large Records

If you are writeing very large records to UniData, you may need to change the records per chunk value in order to accommodate them. This is done by adding the following text in the **Description** box of the UniData Stage Inputs page General tab at job design time:

`RecsPerChunk=`*`nnn`*

Where *nnn* is the number of records returned per chunk. You may need to use trial and error in getting the right value for this. You will get error message when you run the job if you are attempting to return too many records per chunk.

UniData File Meta Data

When importing meta data from UniData 5.2 databases, you use the **Import ➤ Table Definitions ➤ UniData File Definitions** menu item in the DataStage Manager. The resulting **Import Meta Data (Unidata Files)** dialog establishes a connection to a UniData database specified by a host name and a database name. Once connected, a list of accessible files is displayed which includes all F- or LF-type file pointer records in the VOC which are not recognized as UniData system files.

Once a file from this list is selected, the set of visible columns is determined solely from the contents of the file dictionary as follows:

- All D-, I-, and V-type dictionary items are shown including synonyms.

Connecting to UniData 6

The UniData 6 stage can only access UniData 6 accounts that have been made publicly accessible. To connect via DataStage you must then specify a DSN for the UniData 6 account. In order to use the UniData 6 stage, you need a patch release of UniData 6 as follows:

Platform	UniData Server	UniData Client
Windows	6.0.7	6.0B
AIX	6.0.8	6.0B
HP	6.0.9	6.0B
Solaris	6.0.9	6.0B

The stage is not currently supported for Tru64 or Linux.

UniData 6 Account

When a new account is created in UniData 6, it is not a UniData 6 account until MIGRATE.SQL is run in the account. To make an account a UniData 6 account:

1. Telnet into the UniData account as an admin user.
2. At the UniData prompt type MIGRATE.SQL

To make the account accessible:

1. Open the UniAdmin client.
2. Add the account you wish to make accessible to the **Accounts Admin** configuration:
   a. Double click on the **Accounts** item in the tree to open the **Accounts Admin** dialog box.
   b. Click the **New** button to add a new account.
   c. Enter the name of your UniData 6 account and its full pathname.
3. Add the account you wish to make accessible to the **Network Services/Database Admin** configuration:

a Double click the **Database Admin** item under the Network Service branch to open the **Database Configuration** dialog box.

b Click the **Add** button to add a new database.

c Enter a database name, and the full pathname to the UniData 6 account you wish to access.

In a UniData 6 account, all new tables created will be accessible by client applications. Any existing files before MIGRATE.SQL was executed will not be accessible.

To make all existing files accessible:

1. The DataStage 7.0 media contains a UniBasic program called Privilege.B. Copy the file Privilege.B from the DataStage Install Image. This can be found under the following directory structure: *Utilities\Supported\UDT6\Privilege.B*
2. Place this file into the BP directory of the UniData6 account.
3. Telnet to the account using the login credentials of the user that you wish to access these files.
4. Compile and run the program using the following commands.

   BASIC BP Privilege.B
   RUN BP Privilege.B

DataStage DSN for UniData 6 Server

For the UniData 6 stage to connect to a UniData 6 server, a DSN needs to be set up in the *uvodbc.config* file in the DataStage project.

The DSN entry should follow the format:

```
<Name>
DBMSTYPE = UNIDATA
network = TCP/IP
service = udserver
host = IP_ADDRESS:31438
```

Where *IP_ADDRESS* is the IP address of the UniData 6 machine.

An example of a DSN is:

```
<remoteud>
DBMSTYPE = UNIDATA
network = TCP/IP
service = udserver
host = 193.128.90.100:31438
```

UniData 6 Connection Parameters

The UniData 6 connection parameters are set on the **General** and **Details** tabs on the **Stage** page of the UniData 6 stage editor. To connect to a UniData 6 data source:

1. Choose the data source name from the **Data source name** drop-down list box. This drop-down list box contains all the data sources defined under the **Table Definitions ➤ UniData 6** branch in the Repository.

   Note If the data source name you want is not listed, you can either enter the name in the **Data source name** field or define a table definition. For details on how to import or create a table definition, see *"Managing Table Definitions"* in *DataStage Manager Guide*.

   You can also enter a job parameter in this field. For details on how to define and use job parameters, see "Specifying Job Parameters" in *DataStage Designer Guide*.

2. Enter the user name to use in the **User name** field. You can enter a job parameter here, in the form #*parameter*#.
3. Enter the password to use in the **Password** field. You can enter a job parameter here, in the form #*parameter*#, but it will be displayed as asterisks.
4. Enter an optional description of the UniVerse stage in the **Description** field.
5. Click the **Details** tab. The **Details** tab appears at the front of the **Stage** page.
6. Specify the UniVerse account to use for the connection either by account name or account directory:
   - **Use account name**. Enter the account name in the **Account name** field. This must represent a valid UniData 6 account which is also a schema.
   - **Use directory path**. Enter the directory path where the project is stored in the **Directory path** field.

You can use a job parameter in any of the fields described in steps 1, 2, 3, 4, and 6. For details on how to define and use job parameters, see "Specifying Job Parameters" in *DataStage Designer Guide*.

UniData Table Meta Data

When importing meta data from UniData6 databases, you use the **Import ➤ Table Definitions ➤ UniData 6 Table Definitions** menu item in the DataStage Manager. The resulting **Import MetaData (Unidata 6 Tables)** dialog establishes a connection to a UniData database specified by the DSN chosen from the drop-down list of available DSNs .

3

Multivalued Data and Multilevel Files

Multivalued/Multi-Subvalued Data and Associations

Both UniVerse and UniData support multivalued and multi-subvalued data and associations which allow a relationship between a number of multivalued/multi-subvalued fields within a record to be identified. From DataStage 3.1 onwards, meta data imported from UniVerse or UniData includes information about whether a field is single-valued, multivalued, or multi-subvalued, and whether it is part of an association. This is determined as follows in both UniVerse and UniData dictionaries:

- The value of field 6 indicates the field type. S or '' denotes single-valued, M or MV denotes multivalued, and MS denotes multi-subvalued.
- The value of field 7 is the name of the association to which the field belongs.
- A field that is multivalued and does not belong to an association is treated as a member of an association whose name is the same name as the field.

UniVerse dictionaries can also contain Pick-style A- and S-type entries which are interpreted differently:

- The value of field 4 indicates a multivalued attribute. It contains an entry of the form **C;*n*[;*m*]...** for a controlling attribute or **D;*n*** for a dependent attribute. When importing meta data, this is handled like an association with the same name as the controlling attribute. All the controlling and dependent attributes appear as multivalued fields which are members of this association.

- The value of field 5 can be used to override the field type. S or '' denotes single-valued, M or MV denotes multivalued, and MS denotes multi-subvalued.

UniVerse, UniData, and Hashed File stage outputs recognize the presence of multivalued and multi-subvalued column meta data and offer the user the option to normalize the data with respect to a single association generating a separate row of column values for each multivalue in an association. The Un-Normalized options deliver all of the values as a dynamic array containing value marks and subvalue marks.

The Data Browser **Display...** button also offers the Un-normalized (formatted) option, which formats the data in the grid with each value on a separate line in a similar way to RetrieVe or UniQuery.

The following example shows how records containing multivalued data are displayed for a table with six columns: two single-valued, two multivalued and two multi-subvalued. The multivalued and multi-subvalued fields all belong to a single association.

Column Name	Field No.	Field Type	Association
Key	0	S	
Field1	1	S	
Field2	2	MV	Assoc1
Field3	3	MV	Assoc1
Field4	4	MS	Assoc1
Field5	5	MS	Assoc1

The sample records contain the following data:

```
ID:     A
DATA:   BFMCVMGVMNVMRFMDVMHVMOVMSFMEVMISMKSMMVMPVMTSMVFMFVMJSMLVMQVMUSMW
ID:     X
DATA:   YFMFMFMFM
```

The second record is used to illustrate the differences in normalization behavior between UniVerse and UniData when the associated multivalued fields are empty.

The unnormalized view of the table returns the following column values:

Key	Field 1	Field2	Field3	Field4	Field5
A	B	CVMGVMNVMR	DVMHVMMVMS	EVMISMKSMMVMPVMTSMV	FVMJSMLVMQVMUSMW
X	Y				

The unnormalized (formatted) view of the table, only available from the Data Browser, formats the previous row as follows. This is similar to the layout that RetrieVe or UniQuery produce.

Key	Field1	Field2	Field3	Field4	Field5
A	B	C	D	E	F
		G	H	I	J
				K	L
				M	Null
		N	O	P	Q
		R	S	T	U
				V	W
X	Y	Null	Null	Null	Null

The view of the table normalized on Assoc1 will vary slightly between UniData files, UniVerse files, and UniVerse tables.

Key	Field1	Field2	Field3	Field4	Field5
A	B	C	D	E	F
A	B	G	H	I	J
A	B	G	H	K	L
A	B	G	H	M	Null
A	B	N	O	P	Q
A	B	R	S	T	U
A	B	R	S	V	W
X	Y	Null	Null	Null	Null

The result set above will be returned by the UniData stage or the Data Browser during UniData file meta data import, and is equivalent to that returned by the UniData SQL query:

```
SELECT Key,Field1,Field2,Field3,Field4,Field5 FROM Table1 UNNEST
Field2,Field3,Field4,Field5
```

The result set returned by the Hashed File stage or the Data Browser during UniVerse file meta data import differs from the above in that the final row will be omitted:

Key	Field1	Field2	Field3	Field4	Field5
A	B	C	D	E	F

Key	Field1	Field2	Field3	Field4	Field5
A	B	G	H	IsmKsmM	JsmL
A	B	N	O	P	Q
A	B	R	S	TsmV	UsmW

The result set above will be returned by the UniVerse stage or the Data Browser during UniVerse table meta data import, and is equivalent to that returned by the UniVerse SQL query:

```
SELECT Key,Field1,Field2,Field3,Field4,Field5 FROM UNNEST Table1 ON
Assoc1
```

Multiple Data Files (Multilevel Files)

Both UniVerse and UniData support Pick-style multiple data files (multilevel files) which allow multiple subfiles to share a common dictionary. When importing meta data for UniVerse files or tables or UniData files, only a single file or table corresponding to the shared dictionary and default data subfile will be visible and this can be used to import the meta data from the shared dictionary in the same way as for any other files or tables.

In a DataStage job, a specific subfile can be used in a Hashed File or UniData stage input/output by specifying a file name of the form *MFileName,SubFileName*.

The default subfile can also be specified by a file name of the form *MFileName*. This form can also be used in a UniVerse stage input/output.

Index

A
account name
    UniData 6 stage 2–9

C
connection type
    UniData 6 2–9
    UniVerse 2–7
co-resident systems 1–3

D
DSN 1–2, 2–8

I
importing meta data from UniData 5.2 2–6
importing meta data from UniData 6 2–8, 2–10

L
loacl UniVerse data sources 1–1

M
multiple data files (Multilevel files) 3–4
Multivalued/Multi-Subvalued data and associations 3–1

O
ObjectCall 2–1

P
PROTOCOL 2–4

R
remote UniVerse data sources 1–1

U
UDTINI 2–1
UNIAPI.INI 2–1, 2–2, 2–5
UNICLTSTATLEVEL 2–4
UniData 5.1 2–10
UniData 5.2 2–1
UniData 6 2–7
UniData 6 account 2–7
UniData 6 stages
    account name for 2–9
    and UniData 6 connection 2–9
UNILOGDIR 2–3
UNILOGLEVEL 2–3
UNIPACKETSIZE 2–4
UNIRESPONSESIZE 2–3
UNISVRSTATLEVEL 2–4
UniVerse stages
    and UniVerse connection 2–7
UV.ACCOUNT file 1–1
UV/Net 1–1