

IBM[®] DB2[®] OLAP Server[™]



DB2 OLAP Server Installation Guide

Version 8.2

IBM[®] DB2[®] OLAP Server[™]



DB2 OLAP Server Installation Guide

Version 8.2

Note

Before using this information and the product it supports, be sure to read the information in “Notices” on page 117.

First Edition (July 2004)

This edition applies to version 8.2 of DB2 OLAP Server and DB2 OLAP Integration Server (program number 5724-OLP) and to all subsequent releases and modifications until otherwise indicated in new editions.

Portions of this manual © Copyright Hyperion Solutions Corporation 1991, 2004

© **Copyright International Business Machines Corporation 1998, 2004. All rights reserved.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this book. vii

Who should read this book	vii
IBM names for Hyperion products and components	vii
Publications	viii
How to get support	x
Product information	x
Comments on the documentation	xi

Chapter 1. Before you install. 1

Follow the installation checklist	1
Prepare to upgrade DB2 OLAP Integration Server	2
Migrating existing OLAP Metadata Catalogs.	3
Understanding the HYPERION_HOME variable	3
Understanding the ESSLANG variable.	4
Managing ESSLANG settings.	4
About the security file and ESSLANG values	4
Check compatibility with earlier versions.	5

Chapter 2. Hardware and software requirements 7

Server hardware and software requirements	7
Windows server requirements	7
Solaris Operating System server requirements	8
AIX server requirements	9
HP-UX server requirements	10
Linux server requirements	11
Client hardware and software requirements	12
API system hardware and software requirements.	13
Supported ODBC drivers and relational data sources	14
For DB2 OLAP Server	14
For DB2 OLAP Integration Server	15
Supported JDBC drivers for XML Import/Export in OIS	16

Chapter 3. Installing and starting on Windows 19

Installing DB2 OLAP Server	19
Shut down Windows services and applications before installation	20
Installation steps.	20
Security	21
Installing DB2 OLAP Integration Server on Windows	21
Installing DB2 OLAP Server Miner on Windows	22
Installing the runtime client	23
Setting up a silent install	23
Installing the documentation	24
Setting up a silent install of the documentation	24
Starting the Analytic Server on Windows	25
Automating server startup	25
Automating application startup.	26
Stopping DB2 OLAP Server	26
Starting the Integration Server on Windows	26

Starting and stopping Integration Server from the Windows desktop	26
Starting and stopping Integration Server from the DOS command line.	27
Troubleshooting Common Connection Problems	27
Updating the path for Java on Windows (optional)	27
Running DB2 OLAP Server and DB2 OLAP Integration Server as a Windows service.	28
Installing the Analytic Server and Integration Server as Windows services	28
Verifying that the services are running	28
Stopping the services	29
Removing the services.	29
Adding additional software or users later	29
Using agent commands	29
Workflow for implementing external authentication	30

Chapter 4. Installing and starting on UNIX. 33

Installing DB2 OLAP Server on UNIX	33
Pre-installation steps	34
Installation steps for DB2 OLAP Server	34
Installing DB2 OLAP Integration Server on UNIX	35
Installing DB2 OLAP Server Miner on AIX	36
Configuring the environment on UNIX	36
Updating the DB2 OLAP Integration Server environment for ODBC	38
Updating the environment for the database client	38
Enabling use of cache memory locking systems	38
Configuring the cron scheduling daemon	39
Starting the Analytic Server on UNIX.	39
Starting Analytic Server in the foreground	40
Starting the server in the background.	40
Hiding Analytic Server passwords on UNIX	41
Shutting down Analytic Server	41
Starting Integration Server on UNIX	41
Running Integration Server on UNIX Systems.	41
Stopping Integration Server on UNIX Systems.	42
Troubleshooting common connection problems	42
Adding additional software or users later	43
Using agent commands	43

Chapter 5. Configuring DB2 OLAP Server for relational integration 45

Preparing to use SQL Interface or Hybrid Analysis	45
SQL Interface and Hybrid Analysis on Windows	45
SQL Interface and Hybrid Analysis on UNIX	46
What happens during installation on UNIX.	46
Linking SQL Interface to ODBC drivers on UNIX	47
Setting up Teradata on UNIX	47
Completing configuration tasks on UNIX	48
Setting library paths on UNIX	48
Additional tasks.	48
Additional Hybrid Analysis tasks	48
Additional SQL Interface tasks	49

Hybrid Analysis features map 49

Chapter 6. Setting up the sample application for DB2 OLAP Server 51

Loading the sample databases for DB2 OLAP Server 51
Providing user access to the sample applications 52
Preparing the sample aggregate storage application for use 53
Setting the environment for the partitioning sample applications 53
 Creating the partition user 54
 Changing embedded server names in the sample partition definitions 54
Loading data into Samppart and Sampeast 55

Chapter 7. Connecting DB2 OLAP Integration Server 57

Starting Integration Services Console 57
Connecting to server components and data sources 57
 Connecting to Integration Server 58
 Connecting to Analytic Server 59
 Setting connection defaults 60
 Connecting to data sources 60
 Reconnecting when connections are lost 62

Chapter 8. Configuring relational data sources for DB2 OLAP Integration Server 63

About data source configuration 63
 Server and client software and ODBC connections 64
 Supported ODBC drivers 64
Configuring data source names on Windows 64
 Accessing the ODBC Data Source Administrator 65
 Configuring DataDirect Wire Protocol drivers 65
 Testing DataDirect Wire Protocol drivers. 68
 Adding a configuration file entry for DB2 UDB for z/OS and OS/390 68
 Configuring the SQL Server ODBC Driver for Microsoft SQL Server 69
 Configuring the Teradata ODBC driver for Teradata 70
 Editing a Data Source Name. 71
Using Oracle OCI Direct Connection for Oracle 71
Configuring data source names on UNIX systems 72
 Configuring environment variables 72
 Editing the odbc.ini file 72
 Creating DB2 bind packages for DataDirect Wire Protocol driver connections 77
 Adding a configuration file entry for DB2 UDB for z/OS and OS/390 77
After configuration of data source names 78

Chapter 9. Creating, upgrading, and deleting OLAP Metadata Catalogs 79

About OLAP Metadata Catalogs 79
Creating or upgrading OLAP Metadata Catalogs 80
 Creating databases for OLAP Metadata Catalog tables 80

 Creating or upgrading the OLAP Metadata Catalog automatically 80
 Creating the OLAP Metadata Catalog manually 81
 Upgrading the OLAP Metadata Catalog manually 82
Mapping OLAP Metadata Catalogs 84
Deleting OLAP Metadata Catalogs. 84

Chapter 10. Setting up the sample application for DB2 OLAP Integration Server 87

Preparing to set up the standard sample application 88
Setting up the standard sample application automatically 89
Setting up the standard sample application manually 90
Setting up the TBC relational data source 91
 Creating the TBC database 91
 Creating tables for the TBC database 91
 Loading data into the TBC tables 92
Setting up the TBC_MD OLAP Metadata Catalog. 93
 Creating the TBC_MD database. 93
 Creating Tables for the TBC_MD Database 94
 Using the XML Import utility to load data into the TBC_MD tables. 95
 Using XML Import to load data into TBC_MD. 95
Setting up the Hybrid Analysis sample application 96
 Setting up the Hybrid Analysis sample application on DB2 UDB 97
 Setting up the Hybrid Analysis sample application on Oracle 99
 Setting up the Hybrid Analysis sample application on SQL Server 100
 Setting up the Hybrid Analysis sample application on Teradata 102
Setting up the Unicode sample application 104
 Creating the Unicode TBC_U database 104
 Creating the Unicode TBC_MD_U OLAP Metadata Catalog 104
 Setting up the Unicode sample application 105
After setting up the sample application. 106
 Viewing TBC tables and columns. 106
 Viewing OLAP models 106
 Viewing metaoutlines 107

Chapter 11. Upgrading and migrating from earlier versions 109

Migrating DB2 OLAP Server 109
 Supported API Versions 109
 Upgrading applications that use APIs 109
 Compatibility of non-Unicode applications between client and server software 110
 Understanding input/output defaults and upgrading 110
 Upgrading databases to Version 8.2 111
 Moving applications and databases to another computer 112
 Upgrading and moving applications and databases at the same time 113
Migrating DB2 OLAP Integration Server 114
 ODBC drivers on Windows. 114

Supported JDBC drivers for XML import/export 114
Migrating existing OLAP metadata catalogs . . 114
Other notes on migration 115

Notices 117

Trademarks 118

Index 119

About this book

IBM® DB2 OLAP Server is based on products from Hyperion Solutions Corporation, and it includes most of books and online help written by Hyperion. This book was written by IBM and is the starting point for IBM customers. Only the Hyperion books that are relevant to DB2 OLAP Server are included with this product; the exceptions are listed in “Publications” on page viii.

This book and all other documents written by IBM, such as readme files, override and supersede the information in Hyperion documents. If there are contradictions between IBM and Hyperion documentation, the IBM documentation takes precedence. For example, the Hyperion documentation includes references to Hyperion phone numbers and Web sites for getting support for the product. Please disregard them, and instead contact IBM to get support; see How to get support in this chapter for more information.

You will see references to Hyperion and Hyperion’s products in the interface and throughout the Hyperion books. The IBM names for these products are different; see “IBM names for Hyperion products and components” for a comparison.

Who should read this book

Read this book if you are a DB2 OLAP Server administrator who is responsible for:

- Installing and configuring DB2 OLAP Server, including DB2 OLAP Integration Server.
- Designing and creating OLAP applications and multidimensional databases using DB2 OLAP Server.
- Setting up security for DB2 OLAP Server.
- Maintaining OLAP applications and multidimensional databases that are migrated to or created using DB2 OLAP Server.

As the DB2 OLAP Server administrator, you should have experience in networking and system administration, and should understand the analytical requirements of the users who will use the applications created with DB2 OLAP Server.

IBM names for Hyperion products and components

Table 1 lists the IBM names for the Hyperion products and components that are contained in DB2 OLAP Server™ and its features:

Table 1. IBM and Hyperion names

IBM name and abbreviation	Hyperion name and abbreviation
DB2 OLAP Server (no abbreviation)	Essbase Analytic Services (no abbreviation)
DB2 OLAP Integration Server (OIS)	Essbase Integration Services (EIS)
DB2 OLAP Server Administration Services (OAS)	Essbase Administration Services (EAS)
DB2 OLAP Server Deployment Services (ODS) (formerly called DB2 OLAP Server Enterprise Services in Version 8.1)	Essbase Deployment Services (EDS) (formerly called Essbase Enterprise Services, or EES)
DB2 OLAP Server Spreadsheet Services (OSS)	Essbase Spreadsheet Services (ESS)

Table 1. IBM and Hyperion names (continued)

IBM name and abbreviation	Hyperion name and abbreviation
DB2 OLAP Server Miner (no abbreviation)	Not available from Hyperion
Analytic Server	The server component of DB2 OLAP Server
Integration Server	The server component of DB2 OLAP Integration Server

Publications

This section lists books that are included with DB2 OLAP Server and DB2 OLAP Integration Server. These books are now separately installable from their own CD.

After installing the books, you can find them from the **Start > Programs > IBM DB2 OLAP Server 8.2 > Documentation 8.2** information map. The books are stored in the `x:\ibm\db2olap\docs` directory, where `x:\ibm\db2olap` is the directory in which you installed DB2 OLAP Server workstation applications.

Table 2 lists the publications that are included with the DB2 OLAP Server base product.

Table 2. DB2 OLAP Server Publications for DB2 OLAP Server and DB2 OLAP Integration Server

Publication Name	Description
<i>DB2 OLAP Server Installation Guide</i> (SC18-9205)	Explains how to install and configure DB2 OLAP Server and DB2 OLAP Integration Server.
<i>OLAP Miner User's Guide</i> (SC27-1611)	Explains how to use DB2 OLAP Server Miner V8.1.2, a no-charge feature that combines the powerful IBM data mining technologies with OLAP technologies. This is the V8.1 edition of the book; it is the most recent edition and is valid with V8.1.2 of DB2 OLAP Server Miner, which is shipped with V8.2 of DB2 OLAP Server. This book is installed from the DB2 OLAP Server Miner CD; it is not part of the documentation installer CD.
<i>Analytic Services New Features</i> (SC18-9206)	Describes the new functions and features of DB2 OLAP Server.
<i>Spreadsheet Add-in User's Guide for Excel</i> (SC18-9207)	Explains how to use the Spreadsheet Add-in with Microsoft® Excel for Windows® to analyze data.
<i>Analytic Services Database Administrator's Guide</i> (SC18-9208)	Provides administrators with strategies and techniques to implement, design, build, and maintain an optimized multidimensional database. Provides instructions for how to design and build a security system. Provides instructions for how to load, calculate, and report your data.
<i>Analytic Services SQL Interface Guide</i> (SC18-9209)	Explains how to load data from SQL, relational, and flat-file data sources. The SQL Interface tool is part of the Tools Bundle, which is an add-on product for DB2 OLAP Server.
<i>Analytic Services Technical Reference</i> (HTML only)	Provides reference about writing formulas, calculation scripts, report scripts, MaxL statements, and custom-defined macros and functions. It also includes ESSBASE.CFG settings and ESSCMD commands.

Table 2. DB2 OLAP Server Publications for DB2 OLAP Server and DB2 OLAP Integration Server (continued)

Publication Name	Description
<i>Message Reference (HTML only)</i>	Describes common messages in DB2 OLAP Server and their uses and solutions.
<i>Analytic Services API Reference(HTML only)</i>	Provides reference information for functions that are available through the Application Programming Interface, which is part of the DB2 OLAP Server Tools Bundle.
<i>Integration Services New Features (SC18-9210)</i>	Describes the new functions and features of DB2 OLAP Integration Server.
<i>Integration Services System Administrator's Guide (SC18-9211)</i>	Explains how system administrators can configure and maintain DB2 [®] OLAP Integration Server and OLAP Metadata Catalogs, and use the Integration Services Shell.
<i>Integration Services Data Preparation Guide (SC18-9212)</i>	Explains how to prepare relational data for use with the DB2 OLAP Integration Server.
<i>Integration Services ODBC Support Matrix (HTML only)</i>	Lists the Open Database Connectivity (ODBC) and Java [™] Database Connectivity (JDBC) drivers required by the relational database management systems that are supported by DB2 OLAP Integration Server.
<i>Integration Services OLAP Model Data Type Definition (SC18-9213)</i>	Describes the OLAP Model Data Type Definition (DTD) for XML Import/Export .
<i>Integration Services OLAP Metaoutline Data Type Definition (SC18-9214)</i>	Describes the OLAP Metaoutline Data Type Definition (DTD) for XML Import/Export .
<i>DataDirect Connect for ODBC Reference</i>	Provides reference information for the DataDirect Connect for ODBC drivers (formerly called MERANT).

Table 3 lists the books that are available with DB2 OLAP Server Administration Services, DB2 OLAP Server Deployment Services, and DB2 OLAP Server Spreadsheet Services.

Table 3. Publications for other DB2 OLAP Server products

Publication Name	Description
<i>Administration Services Installation Guide (SC18-9215)</i>	Explains how to install and configure DB2 OLAP Server Administration Services.
<i>Administration Services Development Guide (SC18-9276)</i>	Explains how to extend Administration Services and describes the concepts, processes, procedures, formats, tasks, and examples that you need to customize the software.
<i>Deployment Services Installation Guide (SC18-9216)</i>	Explains how to install and configure DB2 OLAP Server Deployment Services.
<i>Spreadsheet Services Installation Guide (SC18-9277)</i>	Explains how to install and configure DB2 OLAP Server Spreadsheet Services.

All the DB2 OLAP Server publications are available online at:

www.ibm.com/software/data/db2/db2olap/library.html

Several IBM Redbooks[™] about DB2 OLAP Server and related topics are available. To view OLAP Redbooks online, visit the IBM Redbook Web site at:

The following books from Version 8.1 have been discontinued in Version 8.2:

- *Quick Path*. In Version 8.2, the closest equivalent to this document is “Hybrid Analysis features map” on page 49, which includes a table that lists where Hybrid Analysis tasks are described.
- *Spreadsheet Add-in Users’s Guide for 1–2–3*. Version 8.2 does not include a spreadsheet add-in for Lotus® 1-2-3®. See “Client hardware and software requirements” on page 12 for more information.
- *MaxL User’s Guide*. This content of this book has been moved to the *Analytic Services Technical Reference*.

The following Essbase Analytic Services and Essbase Integration Services books are not included with this product. The relevant information from them has been added to this book. This book supersedes the following books:

- *Analytic Services Installation Guide*
- *Analytic Services Installation Checklist*
- *Integration Services Installation Guide*
- *Integration Services Installation Checklist*

How to get support

Please disregard all references to Hyperion support in the documentation. Hyperion does not offer support or any services for IBM DB2 OLAP Server, DB2 OLAP Integration Server, or the other DB2 OLAP Server products. In most countries, contact IBM to get support for this product. In some countries, IBM-authorized dealers should contact their dealer support structure instead of the IBM Support Center.

If you have a technical problem, please review and carry out the actions suggested by the product documentation before contacting DB2 OLAP Server Customer Support. This guide suggests information that you can gather to help DB2 OLAP Server Customer Support to serve you better. For information or to order any products, contact an IBM representative at a local branch office or contact any authorized IBM software remarketer. If you live in the U.S.A., you can call one of the following numbers:

- 1-800-237-5511 for customer support
- 1-888-426-4343 to learn about available service options

The Web site for DB2 OLAP Server and its features offers an extensive selection of frequently asked questions (FAQs) and more information, as well as fix packs:

<http://www.ibm.com/software/data/db2/db2olap/support.html>

Product information

If you live in the U.S.A., then you can call one of the following numbers for more information about this product:

- 1-800-IBM-CALL (1-800-426-2255) or 1-800-3IBM-OS2 (1-800-342-6672) to order products or get general information.
- 1-800-879-2755 to order publications.

<http://www.ibm.com/software/data/db2/db2olap/>

The main Web site for DB2 OLAP Server includes links to Fix Packs and support information, announcements, and other product information.

<http://www.ibm.com/software/data/db2/>

The DB2 Product Family Web pages provide current information about news, product descriptions, education schedules, and more.

<http://www.ibm.com/software/info1/websphere/index.jsp?tab=products/appserv>

DB2 OLAP Server works with IBM WebSphere® Application Server, an open-standards transaction engine for dynamic e-business applications.

<http://www.elink.ibm.com/>

Click **Publications** to open the International Publications ordering Web site that provides information about how to order books.

<http://www.ibm.com/education/certify/>

The Professional Certification Program from the IBM Web site provides certification test information for a variety of IBM products.

Comments on the documentation

Your feedback helps IBM to provide quality information. Please send any comments that you have about this book or other DB2 OLAP Server documentation. You can use any of the following methods to provide comments:

- Send your comments using the online readers' comment form at www.ibm.com/software/data/rcf.
- Send your comments by electronic mail (e-mail) to comments@us.ibm.com. Be sure to include the name of the product, the version number of the product, and the name and part number of the book (if applicable). If you are commenting on specific text, please include the location of the text (for example, a title, a table number, or a page number).

Chapter 1. Before you install

This chapter describes the overall installation process and steps to take before installing DB2 OLAP Server V8.2.

The following topics are covered by this chapter:

- “Follow the installation checklist”
- “Prepare to upgrade DB2 OLAP Integration Server” on page 2
- “Understanding the HYPERION_HOME variable” on page 3
- “Understanding the ESSLANG variable” on page 4
- “Check compatibility with earlier versions” on page 5

Follow the installation checklist

This checklist summarizes the major steps of installing DB2 OLAP Server and DB2 OLAP Integration Server:

1. Read about the new features and functions in DB2 OLAP Server and DB2 OLAP Integration Server in *Analytic Services New Features* and *Integration Services New Features*, respectively. You can read these books from our Web site:
<http://www.ibm.com/software/data/db2/db2olap/library.html>
Alternatively, you can install these books and all the documentation for DB2 OLAP Server and DB2 OLAP Integration Server before you install the product. See “Installing the documentation” on page 24 for more information.
2. Verify that you have the hardware and software required by DB2 OLAP Server and DB2 OLAP Integration Server. See Chapter 2, “Hardware and software requirements,” on page 7.
3. Determine what products and features that you purchased and are entitled to install.
4. Determine what free features and options you want to install.
5. If you have an earlier version of DB2 OLAP Server on your computer, prepare for migration:
 - a. Back up your previous installation directory, or at least following files:
 - Your data
 - Essbase.sec
 - license.*
 - b. DB2 OLAP Server Miner does not support migration from V8.1 to V8.1.2, which is the version shipped with DB2 OLAP Server V8.2. If you previously installed V8.1 and want to upgrade to V8.1.2, you must first back up the OLAP Miner data files and configuration files for both the server and clients. The OLAP Miner server configuration file is `omserver.cfg` and the file for the clients is `omclient.cfg`. Then you must uninstall OLAP Miner V8.1 before you can install V8.1.2.

Make sure you understand the issues involved with migration. Read Chapter 11, "Upgrading and migrating from earlier versions," on page 109 before continuing.

- ___ 6. If you have installed an earlier version of this product, stop the Analytic Server and Integration Server, if they are running.
- ___ 7. On Windows, if you have Windows services set up for DB2 OLAP Server and DB2 OLAP Integration Server, stop them and disable them from starting again the next time the system is rebooted. Additionally, shut down all Windows applications.
- ___ 8. Install the server components and client components. See either Chapter 3, "Installing and starting on Windows," on page 19 or Chapter 4, "Installing and starting on UNIX," on page 33.
- ___ 9. Install the documentation. See "Installing the documentation" on page 24.
- ___ 10. If you want to install DB2 OLAP Server Miner V8.1.2, see "Installing DB2 OLAP Server Miner on Windows" on page 22 or "Installing DB2 OLAP Server Miner on AIX" on page 36.
- ___ 11. On UNIX[®] operating systems, configure your environment for both DB2 OLAP Server and DB2 OLAP Integration Server. See "Configuring the environment on UNIX" on page 36.
- ___ 12. If you want to use Hybrid Analysis or the SQL Interface, configure it and set up ODBC using the instructions in Chapter 5, "Configuring DB2 OLAP Server for relational integration," on page 45.
- ___ 13. Set up the DB2 OLAP Server sample applications, if you want to use them. See Chapter 6, "Setting up the sample application for DB2 OLAP Server," on page 51.
- ___ 14. If you installed DB2 OLAP Integration Server:
 - ___ a. Create metadata catalogs. See Chapter 9, "Creating, upgrading, and deleting OLAP Metadata Catalogs," on page 79.
 - ___ b. Configure your relational data sources and set up ODBC by following the instructions in Chapter 8, "Configuring relational data sources for DB2 OLAP Integration Server," on page 63.
 - ___ c. Set up the sample applications. See Chapter 10, "Setting up the sample application for DB2 OLAP Integration Server," on page 87.
- ___ 15. Migrate your applications to Version 8.2, if any. See Chapter 11, "Upgrading and migrating from earlier versions," on page 109.

Prepare to upgrade DB2 OLAP Integration Server

If you are upgrading to DB2 OLAP Integration Server Version 8.2 from a previous release, you should assess several system environment and data migration issues before installing the new software.

The following tasks cannot be completed until after you have installed DB2 OLAP Integration Server:

- On UNIX systems, you cannot update environment variables until after the installation process has completed and the environment variable script files that you need have been created (see "Configuring the environment on UNIX" on page 36).
- On all systems, you cannot migrate existing OLAP Metadata Catalogs until after the installation process has completed and the SQL scripts you need have been created (see "Creating or upgrading OLAP Metadata Catalogs" on page 80).

Migrating existing OLAP Metadata Catalogs

If you have an existing OLAP Metadata Catalog from an earlier version of DB2 OLAP Integration Server, you must upgrade it before using the OLAP Metadata Catalog with the current release of the software.

You cannot upgrade an existing OLAP Metadata Catalog until after you have completed the installation process. The Essbase Integration Services setup program installs the scripts that are needed for the upgrade (see “Creating or upgrading OLAP Metadata Catalogs” on page 80).

CAUTION:

If you have installed the sample application from a previous release of Essbase Integration Services, you should back up your existing sample database, OLAP Metadata Catalog, and the OLAP models and metaoutlines stored in the catalog. You can then upgrade your existing catalog to be compatible with the current release of the software. You cannot, however, store new sample OLAP models and metaoutlines in your previous catalog.

To upgrade an existing OLAP Metadata Catalog, perform the following tasks:

1. Complete the upgrade steps for the existing OLAP Metadata Catalog described in “Creating or upgrading OLAP Metadata Catalogs” on page 80.
2. When connecting to the relational database, make sure to use the same user name and password that you used when you created the original OLAP Metadata Catalog. This procedure adds new tables to the OLAP Metadata Catalog without disturbing the existing information.

CAUTION:

After you upgrade an OLAP Metadata Catalog, you cannot roll back to the previous release. The new release of OLAP Metadata Catalog is not compatible with previous releases of DB2 OLAP Integration Server. In addition, do not attempt to use the Version 8.1 OLAP Metadata Catalog with previous releases of the software; doing so could result in corrupted catalog data.

Understanding the HYPERION_HOME variable

Some internal components are installed in a different directory from the DB2 OLAP Server software. This directory is HYPERION_HOME\common, where HYPERION_HOME is defined as a system variable on Windows and an environment variable on UNIX. If the HYPERION_HOME variable already exists on the computer to which you are installing, the installation program places components common to other OLAP products in this location.

On UNIX, you must specify a location for HYPERION_HOME.

On Windows, the default location for HYPERION_HOME is c:\ibm. Additionally, the installation program sets the ARBORPATH variable as a subdirectory under the HYPERION_HOME directory. For example, if you accept the default value of c:\ibm for HYPERION_HOME, then the default value for ARBORPATH is c:\ibm\db2olap. The only exception is if you install the product directly into the root of your drive, such as c:\; in that case, HYPERION_HOME is set to c:\common.

Understanding the ESSLANG variable

On Windows and UNIX platforms, the installation program prompts you to select a value for the ESSLANG environment variable. ESSLANG points to the correct Global C locale (installed in the ARBORPATH\locale directory). This setting creates or replaces the ESSLANG system environment variable with the selected locale. On UNIX, the ESSLANG setting for a computer must agree with the encoding of the operating system of that computer.

In this release, DB2 OLAP Server requires an ESSLANG value to be set. For information about this new requirement, see the *Database Administrator's Guide*

On Windows and UNIX, at the time of installation, if the ESSLANG system environment variable is already set on the computer to which you are installing the product, that value is selected by default in the installation program. Otherwise, the default value for ESSLANG is English (Latin1). For examples of ESSLANG values for languages other than English, see the list of supported locales in the *Database Administrator's Guide*.

ESSLANG does not need to be set on the computer where DB2 OLAP Server client software is installed, unless it is installed on the same computer as the server components. If you are installing only client software, the installation program does not prompt you to set ESSLANG. When ESSLANG is not set on a client computer, the computer locale is used for encoding.

Managing ESSLANG settings

The ESSLANG environment variable on the server computer must retain the locale value of an application for as long as that application is in use .

If the ESSLANG variable is changed after applications are created on a server, those applications will no longer be able to be opened.

To avoid possible database corruption, set the ESSLANG locale to the same value on the client and the server, in the following circumstances:

- When the client is not Unicode-enabled
- When a Unicode-enabled client works with an outline on DB2 OLAP Server V7.1 FP8 or later (Essbase Release 6.2)
- When a Unicode-enabled client saves an outline over an existing outline on a version of the server that is not Unicode-enabled
- When a Unicode-enabled client saves an outline to a non-Unicode application on a Unicode-enabled server

The ESSLANG locale specifications on clients and server computers can be different when a Unicode-enabled client views and updates an outline belonging to a Unicode-mode application.

On Windows, the installation program does not display the ESSLANG values in full. Instead, it displays only the language and encoding. For example, you see English(Latin1), not English_United States.Latin1@Binary.

About the security file and ESSLANG values

If the ESSLANG system environment variable is already set on the computer to which you are installing DB2 OLAP Server, be sure to choose the same value during installation.

If you are upgrading an existing installation, choosing the wrong ESSLANG value can cause the security file to be corrupted. If the DB2 OLAP Server security file is corrupted during the installation, you can revert to a backup copy by renaming `ESSBASE.SEC.time_stamp` to `ESSBASE.SEC`.

Check compatibility with earlier versions

It is recommended that you install the Version 8.2 OLAP server component with Version 8.2 client products. Client products include the Spreadsheet Add-in for Excel, the Runtime Client, and the Integration Services Console. The Version 8.2 server also works with the Spreadsheet Add-in for 1-2-3 from Version 8.1, Fix Pack 4 (Essbase 6.5.5.3) or later.

There are limitations when you mix different versions of clients and servers in the same environment:

Newer clients with older servers

Using a client that is newer than the server is not recommended or supported.

Older clients with newer servers

Clients at the V8.1 Fix pack 2 or higher level can work with V8.2 servers. Clients earlier than V8.1 Fix Pack 2 might work with the Version 8.2 server, although with feature limitations.

Depending on the service level of an older client, it might not be able to connect with a newer server. Additionally, the newer server might issue messages that are incorrect in the context of the older client.

V8.1 clients can connect only to V8.2 non-Unicode applications.

One client with many servers

If you need to connect a client to several OLAP servers on different machines or operating systems, it is recommended that you make them all the same version and service level. Your client should at least be at the same as the lowest level server. If you experience problems, you might need to use different client machines to match up with the appropriate host, or upgrade all clients and servers to the same service level.

Different versions of clients on the same client system

If you install more than one client component on the same computer, such as Spreadsheet Add-in and Runtime Client, make sure that all of the components are on the same release level. Version 8.1 clients share `.dll` files that may not work with earlier client components.

Do not install V8.1 clients on the same computer as V8.2 clients.

IBM components with Hyperion components

IBM does not support mixing OLAP clients and servers from IBM with OLAP clients and servers from Hyperion Solutions. There are some differences in feature mixes, levels, and some issues in licensing that may cause problems, even though mixing IBM with Hyperion products might appear to work in some situations.

DB2 OLAP Integration Server

DB2 OLAP Integration Server V8.2 is backward compatible with DB2 OLAP Server V8.1, Fix Pack 3 or higher. However, to use the V8.2 Hybrid Analysis features or Unicode, you must connect to DB2 OLAP Server V8.2.

Application Manager from V8.1 (Essbase 6.5.x)

You must install it on a different computer from the computer on which DB2 OLAP Server 8.2 is installed.

Application Manager V8.1 limitations

DB2 OLAP Server Application Manager V8.1 cannot edit non-Unicode V8.2 rules files. Additionally, Application Manager V8.1 does not work with aggregate storage.

Spreadsheet Add-in for Excel, any version earlier than V8.2 (Essbase Analytic Services 7.1)

If you use Spreadsheet Add-in for Excel from a release prior to V8.2, it must be installed on a different computer than the computer on which DB2 OLAP Server V8.2 (Analytic Server from Essbase 7.1) is installed.

Supported API versions

Programs compiled with DB2 OLAP Server V8.2 Application Programming Interface do not work with DB2 OLAP Server V8.1. Additionally, DB2 OLAP Server API programs compiled with header files from a specific release must use the libraries from the same release.

Chapter 2. Hardware and software requirements

This chapter describes the hardware and software that you need to run DB2 OLAP Server and DB2 OLAP Integration Server on various operating systems. For a list of the latest changes and updates to supported software, check the Fix Packs page on our Web site:

<http://www.ibm.com/software/data/db2/db2olap/fixpacks.html>

This chapter contains the following topics:

- “Server hardware and software requirements”
- “Client hardware and software requirements” on page 12
- “API system hardware and software requirements” on page 13
- “Supported ODBC drivers and relational data sources” on page 14
- “Supported JDBC drivers for XML Import/Export in OIS” on page 16

Server hardware and software requirements

Table 4 contains a summary of the system requirements of the server components of DB2 OLAP Server and DB2 OLAP Integration Server:

Table 4. Summary of server requirements

Server Operating Systems	Supported Versions
Microsoft Windows See Table 5 on page 8 for more details.	Pentium [®] or later computers for the following versions of Windows: <ul style="list-style-type: none">• Windows 2000, SP3 or later• Windows 2003• Windows XP (for DB2 OLAP Server Personal Edition, and DB2 OLAP Integration Server, only)
Solaris Operating System See Table 6 on page 8 for more details.	Sun SPARC or ULTRASPARC computers for Sun Solaris 8 and Sun Solaris 9 (Sun OS 5.8 and 5.9)
AIX [®] See Table 7 on page 9 for more details.	PowerPC [®] computers, including RS/6000 [®] for AIX 5L [™] (V5.1 and V5.2). AIX 5L ML1 and ML4 are not supported.
HP-UX See Table 8 on page 10 for more details.	HP PA-RISC for HP-UX 11.0 and 11i On HP-UX, DB2 OLAP Integration Server is a 32-bit application.
Linux [™] See Table 9 on page 11 for more details.	Intel [™] computers for Red Hat Linux AS 2.1

Windows server requirements

DB2 OLAP Server for Windows platforms requires a system that meets specific requirements. These requirements are described in Table 5 on page 8.

Table 5. Windows system requirements

Component	Requirements
Microprocessor	Pentium or higher
RAM	64 MB or greater Member Load formula for DB2 OLAP Integration Server: 10 MB + (700 Bytes x number of members)
Microsoft Windows version	<ul style="list-style-type: none"> • Windows XP (for DB2 OLAP Server Personal Edition, and DB2 OLAP Integration Server, only) • Windows 2003 • Windows 2000, SP3 and later
Disk space	<p>For DB2 OLAP Server, approximately 260 MB for the server software and the sample applications.</p> <p>For DB2 OLAP Integration Server:</p> <ul style="list-style-type: none"> • 105 MB total (without Hybrid Analysis or Unicode sample application) • 125 MB total (after Hybrid Analysis sample data is loaded) • 155 MB total (after Hybrid Analysis and Unicode sample data is loaded) <p>105 to 200 MB for the documentation, depending on your language</p> <p>11 MB for DataDirect ODBC drivers and associated documentation</p>
Network protocol	TCP/IP
Java platform	For DB2 OLAP Server and DB2 OLAP Integration Server: Java Runtime Environment version 1.4.1 (installed by default)

Solaris Operating System server requirements

DB2 OLAP server for the Solaris Operating System requires a system with the minimal hardware and software requirements shown in Table 6.

Table 6. Solaris system requirements

Component	Requirement
Server platform	Sun SPARC or ULTRASPARC machines
Solaris Operating System version	Sun Solaris Operating System 8 and 9 (SUN OS 5.8, and 5.9)
RAM	<p>For DB2 OLAP Server: 128 MB RAM (256 or more MB recommended, especially for large databases)</p> <p>For DB2 OLAP Integration Server:</p> <ul style="list-style-type: none"> • 256 MB for the server software • The minimum ulimit settings are: Stack 50 MB; Data 100 MB; Memory 100 MB • Member Load formula: 50 MB + (700 Bytes x number of members)

Table 6. Solaris system requirements (continued)

Component	Requirement
Swap space	At least two times the amount of physical memory is recommended when physical memory is less than .5 GB. Contact your system administrator or hardware vendor for optimum swap configuration.
Disk space	For DB2 OLAP Server: 306 MB: <ul style="list-style-type: none"> • 188 MB for server software • 118 MB for sample applications For DB2 OLAP Integration Server: <ul style="list-style-type: none"> • 105 MB total (without Hybrid Analysis or Unicode sample application) • 125 MB total (after Hybrid Analysis sample data is loaded) • 155 MB total (after Hybrid Analysis and Unicode sample data is loaded) 11 MB for DataDirect ODBC drivers and associated documentation
Network protocol	TCP/IP (included with Solaris)
Thread type	POSIX Kernel Threads (included with Solaris)
Java platform	For DB2 OLAP Server and DB2 OLAP Integration Server: Java Runtime Environment version 1.4.1 (installed by default)

AIX server requirements

The OLAP server for AIX requires a system with the minimal hardware and software requirements shown in Table 7.

Table 7. AIX system requirements

Component	Requirement
Server platform	RS 6000 PowerPC machines
AIX version	AIX 5L (V5.1 and AIX V5.2), excluding ML1 and ML4
RAM	For DB2 OLAP Server: 128 MB RAM (256 or more MB recommended) Larger databases have greater memory requirements. For DB2 OLAP Integration Server <ul style="list-style-type: none"> • 256 MB for the server software • The minimum ulimit settings are: Stack 50 MB; Data 100 MB; Memory 100 MB • Member Load formula: 50 MB + (700 Bytes x number of members)
Swap space	At least two times the amount of physical memory is recommended when physical memory is less than .5 GB. Contact your system administrator or hardware vendor for optimum swap configuration.

Table 7. AIX system requirements (continued)

Component	Requirement
Disk space	<p>For DB2 OLAP Server: 310 MB:</p> <ul style="list-style-type: none"> • 192 MB for server software • 118 MB for sample applications <p>For DB2 OLAP Integration Server:</p> <ul style="list-style-type: none"> • 105 MB total (without Hybrid Analysis or Unicode sample application) • 125 MB total (after Hybrid Analysis sample data is loaded) • 155 MB total (after Hybrid Analysis and Unicode sample data is loaded) <p>11 MB for DataDirect ODBC drivers and associated documentation</p>
Network protocol	TCP/IP (included with AIX)
Thread type	POSIX Kernel Threads (included with AIX)
Java platform	<p>For DB2 OLAP Server: Java Runtime Environment version 1.4.1</p> <p>For DB2 OLAP Integration Server: Java Runtime Environment version 1.3.1 (installed by default)</p>

HP-UX server requirements

The OLAP server for HP-UX requires a system with the minimal hardware and software requirements shown in Table 8.

Table 8. HP-UX system requirements

Component	Requirement
Server platform	PA-RISC machines
HP-UX version	11.0 and 11i
RAM	<p>For DB2 OLAP Server: 128 MB RAM (256 or more MB recommended) Larger databases have greater memory requirements.</p> <p>For DB2 OLAP Integration Server</p> <ul style="list-style-type: none"> • 256 MB for the server software • The minimum ulimit settings are: Stack 50 MB; Data 100 MB; Memory 100 MB • Member Load formula: 50 MB + (700 Bytes x number of members)
Swap space	At least two times the amount of physical memory is recommended when physical memory is less than 0.5 GB (500 MB). Contact your system administrator or hardware vendor for optimum swap configuration.

Table 8. HP-UX system requirements (continued)

Component	Requirement
Disk space	<p>For DB2 OLAP Server: 665 MB</p> <ul style="list-style-type: none"> • 600 MB for server software • 65 MB for sample applications as shipped <p>For DB2 OLAP Integration Server:</p> <ul style="list-style-type: none"> • 105 MB total (without Hybrid Analysis or Unicode sample application) • 125 MB total (after Hybrid Analysis sample data is loaded) • 155 MB total (after Hybrid Analysis and Unicode sample data is loaded) <p>11 MB for DataDirect ODBC drivers and associated documentation</p>
Network protocol	TCP/IP (included with HP-UX)
Thread type	POSIX Kernel Threads (included with HP-UX)
Java platform	For DB2 OLAP Server and DB2 OLAP Integration Server: Java Runtime Environment version 1.4.1 (installed by default)

Linux server requirements

The OLAP server for Linux requires a system with the minimal hardware and software requirements shown in Table 9.

Table 9. Linux system requirements

Component	Requirement
Server platform	Intel computers
Linux version	Red Hat Enterprise Linux AS 2.1 for Intel Microprocessor based computers
RAM	<p>For DB2 OLAP Server: 128 MB RAM (256 or more MB recommended) Larger databases have greater memory requirements.</p> <p>For DB2 OLAP Integration Server</p> <ul style="list-style-type: none"> • 256 MB for the server software • The minimum ulimit settings are: Stack 50 MB; Data 100 MB; Memory 100 MB • Member Load formula: 50 MB + (700 Bytes x number of members)
Swap space	At least two times the amount of physical memory is recommended when physical memory is less than 0.5 GB (500 MB). Contact your system administrator or hardware vendor for optimum swap configuration.

Table 9. Linux system requirements (continued)

Component	Requirement
Disk space	<p>For DB2 OLAP Server: 375 MB</p> <ul style="list-style-type: none"> • 310 MB for server software • 65 MB for sample applications as shipped <p>For DB2 OLAP Integration Server:</p> <ul style="list-style-type: none"> • 105 MB total (without Hybrid Analysis or Unicode sample application) • 125 MB total (after Hybrid Analysis sample data is loaded) • 155 MB total (after Hybrid Analysis and Unicode sample data is loaded) <p>11 MB for DataDirect ODBC drivers and associated documentation</p>
Network protocol	TCP/IP (included with Linux)
Thread type	POSIX Kernel Threads (included with Linux)
Java platform	For DB2 OLAP Server and DB2 OLAP Integration Server: Java Runtime Environment version 1.4.1 (installed by default)

Client hardware and software requirements

The system requirements of the Runtime Client, the Spreadsheet Add-in for Excel , and the OIS Essbase Integration Services Console are described in Table 10. API client requirements are described in “API system hardware and software requirements” on page 13.

Table 10. Client requirements for DB2 OLAP Server and DB2 OLAP Integration Server:

Component	Requirement
Microsoft Windows version	<ul style="list-style-type: none"> • Windows XP • Windows 2003 • Windows 2000, SP3 and later
Microsoft Excel version	<ul style="list-style-type: none"> • Excel 2003 • Excel 2002 (in Office XP) • Excel 2000 • Excel 97
Lotus 1-2-3 version	Lotus 1-2-3 Version 9.7, for use only with Version 8.1, Fix Pack 4 (Essbase 6.5.5.3) or later of the Spreadsheet Add-in for 1-2-3. Version 8.2 does not include a Spreadsheet Add-in for Lotus 1-2-3 at the 8.2 level.
Microprocessor	Pentium or better or equivalent
Display	Resolution of at least 640 x 480 (800 x 600 or more recommended)

Table 10. Client requirements for DB2 OLAP Server and DB2 OLAP Integration Server: (continued)

Component	Requirement
RAM	For DB2 OLAP Server: <ul style="list-style-type: none"> • 32 MB for the Runtime Client • 16 MB for Spreadsheet Add-in for Excel For DB2 OLAP Integration Server: 64 MB for the OIS Essbase Integration Services Console
Disk space	For DB2 OLAP Server: 31 MB total: <ul style="list-style-type: none"> • 14 MB for the Runtime Client • 17 MB for the Spreadsheet Add-in for Excel For DB2 OLAP Integration Server: 42 MB total <ul style="list-style-type: none"> • 20 MB for console and common files • 22 MB for Java and Jar files
Network protocol	TCP/IP

API system hardware and software requirements

The system requirements of the API client development platforms on Windows and UNIX platforms are described in Table 11.

Table 11. API client development platforms

Platform	Supported versions
Windows version	Windows 2000 SP3 and Windows 2003 on Pentium or better, or equivalent computers
Solaris Operating System version	Sun Solaris Operating System 8 and 9 (SUN OS 5.8 and 5.9) on Sun SPARC or ULTRASPARC computers
AIX version	AIX 5L (V5.1 and V5.2) on PowerPC computers, including RS/6000 Excluding AIX 5L ML1 and ML4
HP-UX version	HP-UX 11.0 and 11i on PA-RISC computers
Linux version	Red Hat Enterprise Linux AS 2.1 for Intel Microprocessor based computers
RAM	<ul style="list-style-type: none"> • 64 MB RAM (128 or more MB recommended) for Windows • Physical memory: 64 MB (256 MB recommended) for AIX, HP-UX, Solaris, and Linux Swap space: At least two times the amount of physical memory is recommended when physical memory is less than .5 GB. Contact your system administrator or hardware vendor for optimum swap configuration.
Disk Space	<ul style="list-style-type: none"> • 15 MB for 32-bit DB2 OLAP Server API on Windows XP and Windows 2000 • 14 MB for AIX, HP-UX, Solaris, and Linux

Supported ODBC drivers and relational data sources

In most cases, DB2 OLAP Server and DB2 OLAP Integration Server require the minimal levels of ODBC drivers. If you install both products, check the following tables to make sure that you use an ODBC driver that meets the minimum requirements of both products.

For DB2 OLAP Server

DataDirect Technologies (formerly MERANT) ODBC driver version 4.2 is provided with the DB2 OLAP Server Version 8.2. DB2 OLAP Server also supports other drivers, which are listed in Table 12.

Table 12. Supported ODBC drivers for DB2 OLAP Server

Relational Database Server (Relational Data Source)	ODBC drivers per supported operating system				
	Windows 2000, Windows 2003, Windows XP	Solaris 8, 9 (Sun OS 5.8, or 5.9)	AIX 5.1, 5.2	HP-UX 11.0, HP-UX 11i	Red Hat Linux AS 2.1
DB2 UDB 7.2 (Wired drivers)	DB2 UDB 7.2 ODBC, DataDirect 4.2	DB2 UDB 7.2 ODBC, DataDirect 4.2	DB2 UDB 7.2 ODBC, DataDirect 4.2	DB2 UDB 7.2 ODBC, DataDirect 4.2	DB2 UDB 7.2 ODBC, DataDirect 4.2
DB2 UDB 8.1 (Wired drivers)	DB2 UDB 8.1 ODBC, DataDirect 4.2	DB2 UDB 8.1 ODBC, DataDirect 4.2	DB2 UDB 8.1 ODBC, DataDirect 4.2	DB2 UDB 8.1 ODBC, DataDirect 4.2	DB2 UDB 8.1 ODBC, DataDirect 4.2
Informix® 9.x ¹	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2
Oracle 8i	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2
Oracle 9i	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2
Teradata V2 R4.1	TUF 7.0 with Teradata 3.2 ODBC	TUF 7.0 with Teradata 3.2 ODBC	TUF 7.0 with Teradata 3.2 ODBC	TUF 7.0 with Teradata 3.2 ODBC	Not Supported
Teradata V2 R5	TUF 7.0 with Teradata 3.2 ODBC	TUF 7.0 with Teradata 3.2 ODBC	TUF 7.0 with Teradata 3.2 ODBC	TUF 7.0 with Teradata 3.2 ODBC	Not Supported
MS SQL Server 7.0	MS SQL Server 3.70 ODBC	DataDirect 4.2	Not supported	DataDirect 4.2	DataDirect 4.2
MS SQL Server 2000	MS SQL Server 3.70 ODBC	DataDirect 4.2	Not supported	DataDirect 4.2	DataDirect 4.2
Sybase 12.0 ¹	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2
Sybase 12.5 ¹ (Adaptive Server)	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2
Access 2000 ¹	Microsoft Driver	Not supported	Not supported	Not supported	Not Supported
Access 2002 ¹	Microsoft Driver	Not supported	Not supported	Not supported	Not Supported

Notes:

1. The Informix 9x, Sybase 12.0, Sybase 12.5, Access 2000, and Access 2002 databases are supported with SQL Interface, but they are not supported with Hybrid Analysis.

For DB2 OLAP Integration Server

DB2 OLAP Integration Server supports the ODBC drivers for the relational database management systems (RDBMS) and operating systems shown in Table 13. Unless otherwise noted, client software is not required for the database server.

Table 13. Supported ODBC drivers for DB2 OLAP Integration Server

Relational Database Server (Relational Data Source)	ODBC drivers per supported operating system				
	AIX 5.1, 5.2	HP-UX 11.0, HP-UX 11i	Solaris Operating System 8, 9 (Sun OS 5.8, 5.9)	Red hat Linux AS 2.1	Windows 2000, Windows 2003, Windows XP
DB2 UDB 7.2, FP7	DB2 UDB 7.2 ODBC, DataDirect 4.2	DB2 UDB 7.2 ODBC, DataDirect 4.2	DB2 UDB 7.2 ODBC, DataDirect 4.2	DB2 UDB 7.2 ODBC, DataDirect 4.2	DB2 UDB 7.2 ODBC, DataDirect 4.2
DB2 UDB 8.1	DB2 UDB 8.1 ODBC, DataDirect 4.2	DB2 UDB 8.1 ODBC, DataDirect 4.2	DB2 UDB 8.1 ODBC, DataDirect 4.2	DB2 UDB 8.1 ODBC, DataDirect 4.2	DB2 UDB 8.1 ODBC, DataDirect 4.2
DB2 UDB for z/OS® and OS/390® V7	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2
Informix 9.2	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2
Informix 9.3	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2
MS SQL Server 7.0	Not supported	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2	MS SQL Server 3.7 ODBC (included in MDAC 2.6)
MS SQL Server 2000	Not supported	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2	MS SQL Server 3.7 ODBC (included in MDAC 2.6)
Oracle 8i (Oracle 8i 32-bit client required with OCI direct connection)	DataDirect 4.2 Wire Protocol, OCI direct connection	DataDirect 4.2 Wire Protocol, OCI direct connection	DataDirect 4.2 Wire Protocol, OCI direct connection	DataDirect 4.2	DataDirect 4.2 Wire Protocol, OCI direct connection
Oracle 9i (Oracle 8i 32-bit client required with OCI direct connection)	DataDirect 4.2 Wire Protocol, OCI direct connection	DataDirect 4.2 Wire Protocol, OCI direct connection	DataDirect 4.2 Wire Protocol, OCI direct connection	DataDirect 4.2	DataDirect 4.2 Wire Protocol, OCI direct connection

Table 13. Supported ODBC drivers for DB2 OLAP Integration Server (continued)

Sybase 12.0	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2
Sybase 12.5	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2	DataDirect 4.2
Teradata V2 R5.0	TUF 7.0 with Teradata 3.2 ODBC	TUF 7.0 with Teradata 3.2 ODBC	TUF 7.0 with Teradata 3.2 ODBC	Not Supported	TUF 7.0 with Teradata 3.2 ODBC

ODBC drivers used with DB2 OLAP Integration Server must be thread-safe. All drivers listed in Table 13 on page 15 are thread-safe. HP-UX requires kernel-threads. Solaris Operating System requires Solaris threads, and AIX requires kernel-threads with pthread API.

Supported JDBC drivers for XML Import/Export in OIS

DB2 OLAP Integration Server requires JDBC drivers to support the XML Import/Export function.

- JDBC Type 1 is installed when you install the Java Runtime Environment. It is compatible with all RDBMSs supported by DB2 OLAP Integration Server, with the following exceptions for Teradata.
 - The JDBC Type 1 driver is not officially supported by Teradata; however, it works on all platforms except HP-UX.
 - When Teradata is run on the HP-UX platform, the JDBC Type 3 driver is required.

Teradata ODBC drivers must be purchased separately from the NCR Corporation.

- The JDBC Type 3 is supported by Teradata driver works on all platforms.
- The JDBC Type 4 driver is required by DB2 OLAP Integration Server if you are using Oracle with OCI. The JDBC Type 4 driver for Oracle works on all platforms.

Table 14. Supported JDBC drivers for DB2 OLAP Integration Server

JDBC Type	Supported JDBC Drivers for Oracle and Teradata per Operating System			
	AIX 5.1, 5.2	HP-UX 11.0, HP-UX 11i	Solaris Operating System 8, 9 (Sun OS 5.8, 5.9)	Windows 2000, Windows XP
For Oracle:				
JDBC Type 1	Supported	Supported	Supported	Supported
JDBC Type 4 ¹	Supported	Supported	Supported	Supported
For Teradata:				
JDBC Type 1 ²	Not supported	Not supported	Not supported	Not supported
JDBC Type 3 ³	Supported	Supported	Supported	Supported

Notes:

1. JDBC Type 4 drivers must be implemented when Oracle with OCI is being used.
2. For Teradata, JDBC Type 1 drivers function for all platforms except HP-UX, but are not supported by NCR Corporation.
3. JDBC Type 3 drivers must be purchased separately for Teradata from the NCR Corporation.

Chapter 3. Installing and starting on Windows

This chapter tells you how to install, configure, and start components of DB2 OLAP Server Version 8.2 on Windows operating systems.

The following topics are covered in this chapter:

- “Installing DB2 OLAP Server”
- “Installing DB2 OLAP Integration Server on Windows” on page 21
- “Installing DB2 OLAP Server Miner on Windows” on page 22
- “Installing the runtime client” on page 23
- “Setting up a silent install” on page 23
- “Installing the documentation” on page 24
- “Starting the Analytic Server on Windows” on page 25
- “Starting the Integration Server on Windows” on page 26
- “Troubleshooting Common Connection Problems” on page 27
- “Running DB2 OLAP Server and DB2 OLAP Integration Server as a Windows service” on page 28
- “Adding additional software or users later” on page 29
- “Updating the path for Java on Windows (optional)” on page 27
- “Using agent commands” on page 29
- “Workflow for implementing external authentication” on page 30

Installing DB2 OLAP Server

Make sure you consider the following issues about how DB2 OLAP Server is installed in version 8.2.

Upgrading from version 8.1 to version 8.2 on the same Windows computer

If you have V8.1 of DB2 OLAP Server on your Windows computer, and you want to install V8.2 on that computer, it is recommended that you make a copy of your existing OLAP directory. Then install V8.2 (in a directory whose name does not contain blanks, underscores, dashes, or other special characters) and import your application files and data into the new version.

You can install V8.2 on top of V8.1, and your existing OLAP applications will be available to the new or upgraded version. But when you install a new version to the same directory, you should take the following steps as a precaution:

- Back up the following files:
 - Your data
 - Essbase.sec
 - license.*
- Stop all OLAP services. See “Shut down Windows services and applications before installation” on page 20 for instructions.

Make sure you are aware of all migration issues before starting the installation. See Chapter 11, “Upgrading and migrating from earlier versions,” on page 109.

Upgrading from version 7.1 to version 8.2 on the same Windows computer

If you want to install V8.2 and you currently have V7.1 installed, you must first migrate to V8.1 before installing and migrating to V8.2. For instructions, see the V8.1 edition of the *DB2 OLAP Server Installation Guide*, available from our Web site:

<http://www.ibm.com/software/data/db2/db2olap/library.html>

Do not install in a directory whose name includes blanks

If you install in a directory whose name contains blanks, some functions might fail. Additionally, other functions might produce unpredictable results if the name of the installation directory contains underscores, dashes, or other special characters.

If you have already installed DB2 OLAP Server Administration Services V8.2

If you have installed DB2 OLAP Server Administration Services V8.2, make sure you reboot your computer before starting the installation of DB2 OLAP Server or DB2 OLAP Integration Server.

Do not reinstall V8.1 on top of V8.2

After you install V8.2, if for any reason you want to reinstall V8.1 on the same machine, do not install it in the same directory. For example, if you installed V8.2 in `c:\ibm\db2olap`, you must install 8.1 in another directory.

Shut down Windows services and applications before installation

Before installing on Windows, if you have Windows services set up for a previous installation of DB2 OLAP Server and DB2 OLAP Integration Server, stop them and disable them from starting again the next time the system is rebooted.

For example, on Windows 2000, to stop and disable the Windows services:

1. Select **Start > Programs > Administrative Tools > Services**.
2. In the Services window, double click **IBM DB2 OLAP Server** to display the Properties window.
3. From the Startup type list, select **Disabled**.
4. Click **Stop**.
5. Click **OK**.
6. Repeat this process for **IBM DB2 OLAP Integration Server** if it is installed.

Additionally, shut down all Windows applications before starting the installation.

Installation steps

To install and set up DB2 OLAP Server on Windows:

1. Stop the Windows services for DB2 OLAP Server and DB2 OLAP Integration Server, if they are running.
2. Stop all other Windows programs, if any are running.
3. Insert the DB2 OLAP Server CD into the CD-ROM drive to start the installation program. If the program does not start automatically, you can run `setup.exe` from the root directory.
4. Click **Next** on the Setup window. To cancel installation at any time, click **Cancel**. To display Help at any time, press F1.
5. Press **Accept** to accept the Software License Agreement.
6. On the Select Installation Options window, click either **Server** or **Client** or both, and click **Next**.

7. On the Select Locale screen, select the language you want and click **Next**.
8. When prompted, select the edition of DB2 OLAP Server that you purchased (Standard, Personal, or Developer), and click **Next**.
9. On the Select Options Purchased screen, select the optional components that you purchased. This panel might be filled in if you are migrating from a previous OLAP Server release and the previous license information can be read. If you purchased the Standard Edition or Developer's Edition, you are prompted to enter the number of concurrent licenses you purchased. Then click **Next**.
10. In the Choose Destination Location panel, enter the drive and directory where you would like to install DB2 OLAP Server, and click **Next**. IBM strongly recommends that you install DB2 OLAP Server and its features into a new separate directory; do not install into directory in which a product other than DB2 OLAP Server and its features has been installed. Additionally, do not install in a directory whose name includes blanks.
11. Select the type of installation that you want: **Typical** or **Custom**.
12. If you selected a Custom installation, the Select Components window displays. Check the components you want and click **Next**.
13. Enter a name for the folder into which DB2 OLAP Server is installed, or accept the default. Click **Next**.
14. Click **Next** to start copying the files to your computer. When the installation program has finished, restart your computer.

Security

The `essbase.sec` file stores information about users, passwords for native security, groups, privileges, and applications. Each time that you successfully start the server component, `essbase.exe`, DB2 OLAP Server makes a backup copy of the security file (`essbase.bak`). If you attempt to start the Agent and a password prompt is not displayed or your password is rejected, no backup file is created. You can restore from the last successful startup by copying `essbase.bak` to `essbase.sec`. Both files are in the `bin` directory where you installed the server component.

Installing DB2 OLAP Integration Server on Windows

DB2 OLAP Integration Server is an add-on feature of DB2 OLAP Server. You can install it when you install DB2 OLAP Server, or you can install it separately.

To install the DB2 OLAP Integration Server separately:

1. Get ready to install. Verify that you have the necessary software product components and that the server and client computer systems on which you will install DB2 OLAP Integration Server meet system requirements.
2. Check for any migration issues.
3. Install DB2 OLAP Server if you have not already done so.
4. Install the server component of DB2 OLAP Integration Server from the DB2 OLAP Server CD-ROM for your operating system.
5. Make sure you have Java Runtime Environment set up on your system. See "Updating the path for Java on Windows (optional)" on page 27 for more information.
6. Install the client component (the Essbase Integration Services Console) from the DB2 OLAP Server CD-ROM for Windows.

7. Configure a data source using ODBC. See Chapter 8, “Configuring relational data sources for DB2 OLAP Integration Server,” on page 63. On the computer running DB2 OLAP Integration Server, create Open Database Connectivity (ODBC) connections to an OLAP Metadata Catalog data source and to the relational data source that you use for creating OLAP models and metaoutlines.
8. Create an OLAP Metadata catalog. See Chapter 9, “Creating, upgrading, and deleting OLAP Metadata Catalogs,” on page 79 for more information.
9. Set up the sample applications. See Chapter 10, “Setting up the sample application for DB2 OLAP Integration Server,” on page 87.

Installing DB2 OLAP Server Miner on Windows

DB2 OLAP Server V8.2 includes DB2 OLAP Server Miner 8.1.2, also called OLAP Miner, which is a no-charge feature of DB2 OLAP Server that brings together IBM’s data mining and OLAP technologies. OLAP Miner applies a sophisticated data mining algorithm called Deviation Detection to the data stored in OLAP cubes. The algorithm scans slices of cubes, identifying values that deviate from the norm. The results of a data mining run can be used by the business analyst to focus on the interesting or surprising values in context of the source OLAP cube. You can still use the usual slice, dice, and drill operations to explore the data, but now you let the values discovered by OLAP Miner drive your analysis. See the *OLAP Miner User’s Guide* for information about what OLAP Miner can do for you.

DB2 OLAP Server Miner V8.1.2, requires the DB2 OLAP Server V8.2 API to be installed on the computer on which you install the OLAP Miner server. Java is a requirement for both DB2 OLAP Server and OLAP Miner.

DB2 OLAP Server Miner does not support migration from V8.1 to V8.1.2. If you previously installed V8.1 and want to upgrade to V8.1.2, you must first uninstall V8.1.

OLAP Miner has a client component that runs on Windows and a server component that runs on both Windows and AIX. For information about installing the OLAP Miner server component on AIX, see “Installing DB2 OLAP Server Miner on AIX” on page 36.

On Windows, you can install the OLAP Miner client on the same computer as the OLAP Miner server or on another computer.

To install OLAP Miner on Windows:

1. Insert the DB2 OLAP Server Miner installation CD into your CD-ROM drive.
2. Run the setup.exe file to begin the installation program.
3. The installation program prompts you for the following information:
 - If you are installing the OLAP Miner server, you need
 - The alias name of the OLAP server you will examine with OLAP Miner.
 - The host address (or IP address) of the server component of DB2 OLAP Server. The address is added to the bin\OMServer.cfg file.
 - If you are installing the OLAP Miner client, you need the host address (or IP address) of the computer on which the OLAP Miner server is installed. This information is stored in the bin\OMClient.cfg file.

Installing the runtime client

When you install any components on Windows, you also install the runtime client by default. You can also install only the runtime client, without installing any other components.

To install the runtime client by itself:

1. Launch the Windows install program.
2. On the Select Installation Options window, select **Client** only.
3. On the Select Locale window, select the language you want and click **Next**.
4. Leave all the check boxes empty on the Select Installation Options window and click **Next**.
5. On the Choose Destination Location window, enter the drive and directory where you would like to install the DB2 OLAP Server runtime client and click **Next**. IBM strongly recommends that you install the runtime into a new separate directory. Do not install into directory in which a product other than DB2 OLAP Server and its features has been installed.
6. Select **Custom** on the Select Installation Type window.
7. On the Select Components window, if you want to install only the runtime client, deselect all the checked components except for Runtime Client. Click **Next**.
8. Enter a name for the folder into which the runtime client is installed, or accept the default, on the Select Folder window. Click **Next**.
9. Click **Next** to start copying the files to your computer. You might need to restart your computer after the files are copied.

Setting up a silent install

On Windows, you can set up a silent installation (also called an unattended install) so that your users do not have to type responses to the prompts of the installation program. You can set up a silent installation for the following components:

- DB2 OLAP Server, including the Runtime Client
- DB2 OLAP Integration Server
- DB2 OLAP Miner

You can also set up a silent installation for the documentation; see “Setting up a silent install of the documentation” on page 24.

To perform a silent install, take the following steps:

1. Record your responses to the installation program prompts. To record your responses, run the following command:

```
setup.exe -r
```

This will record your responses to the installation prompts into a file called `setup.iss`, which is stored in `c:\WINNT`. If you select Reboot at the finish, the reboot will be recorded, and subsequent silent installations will also reboot.
2. Copy the `setup.iss` file into a temporary directory on the computer on which you are installing the product.
3. To play the install back silently on the target machine, run the `setup.exe` command using the `-s` parameter (for silent installation) and the `-f1` parameter (which indicates the location of the `setup.iss` file). For example, if you stored the `setup.iss` file in the `C:\TEMP` directory, enter the following command:

```
setup.exe -s -f1C:\TEMP\setup.iss
```

Installing the documentation

The books for DB2 OLAP Server, DB2 OLAP Integration Server, and DB2 OLAP Sever Miner are installable separately, on Windows only, from a separate CD. To start the installation, launch the DB2 OLAP Server Documentation CD, or run the **setupwin32.exe** file in the root of the CD.

If you have older versions of the documentation you want to keep, select an installation directory other than %arborpath%\docs.

If you select a Typical installation, all of the documentation is installed. If you want only a subset of the documentation, choose the Custom option and deselect the books you do not want installed.

Setting up a silent install of the documentation

To run a silent installation for the documentation, take the following steps:

1. Ensure that the Java Runtime Environment 1.4.1 is installed on the computer and is available in the system path.
2. Load the documentation CD in your CD drive.
3. Perform one of the following actions:
 - If you accept the default settings in the installation program, it installs all the documentation, in English only. If you want to install all the English documentation, run the following command from a command prompt:

```
java -cp d:\setup.jar run -silent
```

Where *d*: is your CD drive.

- If you want to install documentation from a language other than English, or install only a subset of books, you must customize your silent installation with the OLAPDoc.ini file. The OLAPDoc.ini file is stored on the documentation CD and contains values that override the default settings in the installation program. Copy it to a temporary directory and update the values according to the directions in the file. Then run the following command:

```
java -cp d:\setup.jar run -silent -options c:\temp\OLAPDoc.ini
```

Where *d*: is your CD drive and *c:\temp* is the temporary directory in which you updated the OLAPDoc.ini file.

Depending on what else is installed on the computer, the documentation is installed in one of the following directories:

\$ARBORPATH\docs

If DB2 OLAP Server V8.2 is installed on the computer, the V8.2 documentation is installed here.

\$ARBORPATH\docs82

If the documentation for DB2 OLAP Server V8.1 is installed on the computer, the V8.2 documentation is installed here.

c:\ibm\db2olap\docs

If DB2 OLAP Server is not installed on the computer, the V8.2 documentation is installed here.

Starting the Analytic Server on Windows

DB2 OLAP Server is automatically started as a service on your workstation. If you should need to start it manually by using the command line, follow these steps:

1. At the command prompt, enter:

```
Essbase
```

If an error occurs during startup, check your path and environment settings. If your path and environment settings are correct, ensure that the appropriate communications protocol is installed and running on your Windows workstation.

2. If this is a first-time installation, DB2 OLAP Server prompts you to enter the following information:

Company name

The company name that you want to use in your server license registration.

Your name

The name to use as the DB2 OLAP Server supervisor ID. Use this name for your initial login from a client to the server.

System password

This password is required whenever you start DB2 OLAP Server. It is also required to gain access to the server from clients when you use the supervisor account.

Verification of the correctness of the information you entered

Enter Y to verify the entries, or N to retype your entries.

You are asked to confirm your choices. Type 1 to verify your entries or 2 to retype them. When you enter Y, the information will be saved. At subsequent logins, you will be prompted for the system password only.

3. If this is not a first-time installation, DB2 OLAP Server prompts you for the system password that you specified when you first ran DB2 OLAP Server.

DB2 OLAP Server displays this message when the server is started and ready for use:

```
Waiting for Client Requests. . .
```

You can now use the Agent window to perform high-level supervisor operations. For a list of available Agent commands, see “Using agent commands” on page 29.

After you start the Agent, all Agent activities are recorded in the log (essbase.log), which is stored in the ARBORPATH directory. After you start individual applications, application and database activities are recorded in an application log (appname.log), which is stored in the application directory (ARBORPATH\app\appname). For more information, see “Using Analytic Services Logs” in the *Database Administrator’s Guide*.

For more information about starting and stopping Analytic Server, see “Running Analytic Servers, Applications, and Databases” in the *Database Administrator’s Guide*.

Automating server startup

On Windows, you can place the OLAP server component in a startup folder so it will start automatically when Windows boots. You can also specify the server password, to avoid having to type it each time the OLAP Server starts.

To automate the OLAP server startup on Windows platforms:

1. Find the Windows Start Menu customization controls.
For example, on Windows 2000, select **Start > Settings > Taskbar & Start Menu Advanced** tab.
2. Click **Add** and specify the location of the Agent program file, `essbase.exe`.
If you installed to the default directory, this is `c:\ibm\db2olap\bin\essbase.exe`. If you want the DB2 OLAP Server system password to be automatically entered each time DB2 OLAP Server starts, type the password after `essbase.exe` on the command line. For example:

```
c:\ibm\db2olap\bin\essbase.exe password
```


where `password` is your DB2 OLAP Server system password.
3. Click **Next**, and then double-click the **StartUp** or **Startup** folder.
4. Type `Essbase`, or your server name, and click **Finish**.

Automating application startup

You can automatically start up an application or database. When the OLAP server is started, DB2 OLAP Server starts any application or database that is defined with an automatic startup setting. You define startup settings in Administration Services (Application Properties window and Database Properties window).

Stopping DB2 OLAP Server

To stop the server from any workstation that is connected to it, use one of the following two methods:

- For DB2 OLAP Server, you can use Windows Services:
 1. Click **Start** → **Settings** → **Control Panel** → **Services**
 2. On the Services window, select the `Essbase` server.
 3. Click **Stop**.
- For DB2 OLAP Server, you can use the `SHUTDOWNSERVER` command in `ESSCMD`. See the *Database Administrator's Guide*, and the `ESSCMD` online help for more information.

Starting the Integration Server on Windows

After you install the server software on Windows systems, you can choose either of the following options for Integration Server operation:

- “Starting and stopping Integration Server from the Windows desktop”
- “Starting and stopping Integration Server from the DOS command line” on page 27
- “Running DB2 OLAP Server and DB2 OLAP Integration Server as a Windows service” on page 28

Before starting Integration Server as a Windows service, you must first enable the Windows service. See “Running DB2 OLAP Server and DB2 OLAP Integration Server as a Windows service” on page 28 for more information.

Starting and stopping Integration Server from the Windows desktop

To start Integration Server from the Windows desktop Start menu:

1. On the Windows desktop, click **Start**.

2. Select **Programs** → **IBM DB2 OLAP Server 8.2** → **IBM DB2 OLAP Integration Server** → **Integration Server** to display the Integration Server window.

To stop Integration Server from the Windows desktop:

1. In the Windows task bar, right-click the button representing Integration Server.
2. From the pop-up menu, select **Close**.

Starting and stopping Integration Server from the DOS command line

To start Integration Server from the command line, start a DOS window and type `startup.bat` at the command prompt.

When you first start Integration Server, it creates the Integration Server log, `olapisvr.log`, in the log directory. To write to a log in another directory, add an entry to the `startup.bat` file or the `eis.cfg` file specifying the directory. Here is the syntax for the `eis.cfg` file:

```
[-E]=directory_name\log_name
```

Do not type the `.log` file extension. The `.log` extension is automatically appended to the name that you type for the log.

To stop Integration Server from the command line, close the Integration Server window, or use Integration Services Shell.

For more information on starting and stopping Integration Server from the command line using the `startup.bat` command, and on using the `startup.bat` file or `eis.cfg` file to store startup parameters, see the *Integration Services System Administrator's Guide*.

Troubleshooting Common Connection Problems

Before attempting a client-server connection, test the protocol installation to ensure that the connection is installed properly.

Use the PING command on the server to check that it is running and visible to the client. If the PING command is successful but the TELNET command is not, then there may be a problem with the `inet` daemon. If the PING command is unsuccessful on the server, you might have a routing or hardware problem.

Updating the path for Java on Windows (optional)

On Windows, the installation program automatically installs and configures the correct level of Java Runtime Environment (JRE 1.4.1) required by DB2 OLAP Server and DB2 OLAP Integration Server, so no more action is needed. The JRE is required to support features such as custom-defined macros and custom-defined functions.

However, the JRE is set up so that only DB2 OLAP Server and DB2 OLAP Integration Server can use it. No other program can access that JRE unless you update the `PATH` system variable.

To allow other applications to access the JRE that is installed with DB2 OLAP Server, you need to add this variable to your `PATH` statement:

%JREHOME%\bin

For example, you can set your PATH to the following value:

```
PATH=%JREHOME%\bin;%PATH%
```

After the Java components are installed on your system, you can use the Java Development Kit (JDK) installed with this product to create your own custom-defined macros and custom-defined functions.

Running DB2 OLAP Server and DB2 OLAP Integration Server as a Windows service

The server components of DB2 OLAP Server and DB2 OLAP Integration Server can run as Windows services. Running these products as services lets you start and stop them from the Windows Control Panel. When running as automatic services, the Analytic and Integration Servers start in the background when you start Windows.

The Windows Task Manager does not enable you to terminate any process started as a service. If the servers are started as a service, and if it or the server freezes, you must shut down the server computer to stop the service.

Installing the Analytic Server and Integration Server as Windows services

To install the Analytic Server and Integration Server or both as services that start automatically when you start Windows, open a command prompt and type one or both of the following commands:

For DB2 OLAP Server

```
essbase.exe -service install
```

For DB2 OLAP Integration Server

```
olapisvc -i
```

A message is displayed in the command prompt window that the service is installed. The Agent is now listed as `essbase service` in the Services dialog box. The Agent is now set to start automatically when you start Windows.

To run the Analytic Server and Integration Server as Windows services, the operating system needs to know where servers is located. Before running the servers as an automatic service, make sure that `%ARBORPATH%\BIN` has been added to the system-wide path variable.

Verifying that the services are running

To verify that the Analytic Server and Integration Server running as automatic Windows services after you reboot:

1. Reboot the Windows server.
2. Select **Start** → **Settings** → **Control Panel**.
3. Open the Services dialog box and make sure that IBM DB2 OLAP Server and IBM DB2 OLAP Integration Server are in the list of services, that the **Startup Type** is Automatic, and that the **Status** is Started.

Stopping the services

The servers are listed as IBM DB2 OLAP Server and IBM DB2 OLAP Integration Server in the Windows Services dialog box.

To stop the services, select **IBM DB2 OLAP Server**, right click, and select **stop**. Repeat the process with **IBM DB2 OLAP Integration Server**.

Stopping the services can take some time, depending on how many application servers are running. The time that it takes is the same amount it would take to stop the servers if they were running from the command line or as background processes.

Removing the services

To remove the services:

1. Stop the services.
2. From a command prompt, type one or both of the following commands:

For DB2 OLAP Server

```
essbase.exe -service remove
```

For DB2 OLAP Integration Server

```
olapisvc -u
```

A message is displayed in the command prompt window that the services are removed.

Removing a service only removes it from the list of Windows registered services. It does not uninstall DB2 OLAP Server or DB2 OLAP Integration Server.

Adding additional software or users later

The DB2 OLAP Server installation program prompts you for the edition and options that you have purchased, and uses that information to enable DB2 OLAP Server and add-on components.

If you install a license of DB2 OLAP Server and later want to add more features or additional users, you use one of the following methods:

- Select **Modify Component or License Information** from the **Start → Programs → IBM DB2 OLAP Server 8.2** menu.
- Run the **SETUP.EXE** program either from the root directory or from the CD-ROM.
- Use the Windows Add/Remove Software utility, which is available from the Control Panel.

Using agent commands

DB2 OLAP Server has a server process called the Agent. When you start the main server program (ESSBASE), the Agent becomes the active terminal, unless you start the server in the background. The Agent acts as a console for high-level server operations. Pressing the Enter key in the Agent terminal provides a list of all available commands.

Note: The Agent commands are available only when the server is started in the foreground.

The following table shows the commands that are used to manage DB2 OLAP Server.

When you start DB2 OLAP Server, the Agent window opens. The Agent window is the console for high-level server operations. Press **Enter** to view the following list of all available commands.

Table 15. Agent Commands

Command	Description
START <i>appname</i>	Starts the specified application.
STOP <i>appname</i>	Stops the specified application.
USERS	Displays a list of all users that are connected to the server, the total number of connections, and the number of ports that are available.
LOGOUTUSER <i>username</i>	Disconnects a user from the server and frees a port. This command requires the DB2 OLAP Server password.
PASSWORD	Changes the system password that is required to start DB2 OLAP Server.
VERSION	Displays the server version number.
HELP	Lists all valid commands and their respective functions.
PORTS	Displays the number of ports that are installed on the OLAP server and how many are in use. Number of available ports = number of licensed ports + 1 system administrator port.
DUMP <i>filename</i>	Dumps information from the DB2 OLAP Server security system to a specified file in ASCII format. If you do not supply a path with the file name, the file is saved to the \ARBORPATH\bin directory. This command requires the DB2 OLAP Server password.
QUIT/EXIT	Shuts down all open applications and stops DB2 OLAP Server.

Workflow for implementing external authentication

This section helps you set up DB2 OLAP Server to use external authentication in addition to, or in place of, the DB2 OLAP Server security system. Using external authentication to manage user logins provides two main benefits:

- The existing corporate structure of user accounts can be employed by Analytic Services and other OLAP products, reducing administrative overhead.
- The benefit of single sign-on between DB2 OLAP Server and other OLAP products is added, eliminating the need for users to log on multiple times with multiple user names and passwords.

The following workflow describes the steps that are required to implement external authentication and single sign-on with DB2 OLAP Server:

1. Decide which of the supported authentication providers, on which platforms, to make available in the security realm. See Table 20.

Table 16. Supported authentication providers and platforms

Operating system	Lightweight Directory Access Protocol (LDAP) V3 compatible directories	NT LAN Manager (NTLM)	Microsoft Active Directory 2000 and 2003
Windows 2000 Server and Advanced Server, SP3 or later	X	X	X
Windows 2003	X	X	X
UNIX	X	X Requires installation of Remote Authentication Module	X

Note: The tested and supported LDAP servers are: iPlanet 5.2, Novell eDirectory 8.7, IBM Tivoli® Directory Server V5.1, and IBM Lotus Domino® LDAP 5.x and 6.0. iPlanet is now known as Sun Open Net Environment (Sun ONE).

2. The installation program for DB2 OLAP Server sets the environment variables you need. To verify the settings for your operating system, see Security Platform Reference section of the *Analytic Services Technical Reference*, under *Java Version Considerations* and *Setting Environment Variables for Java*.
3. If you are using an NT LAN Manager provider and Analytic Server runs on a Windows platform, the account that runs Analytic Server must be a domain user rather than a local Windows user. Additionally, you must grant the following user rights to the account that runs Analytic Server:
 - Access this computer from the network
 - Act as part of the operating system

For instructions on granting the above rights, see the Security Platform Reference section of the *Analytic Services Technical Reference*, under *NT LAN Manager Notes*.

4. If you are implementing security using an NT LAN Manager provider and are using a UNIX platform for DB2 OLAP Server, DB2 OLAP Server Administration Services, or both, ensure that the Remote Authentication Module is installed on a separate Windows 2000 or Windows 2003 server. You can install the Remote Authentication Module by running the ram\setup.exe file from the DB2 OLAP Server CD. After the Remote Authentication Module is installed, you must also provide its URL as a value to the <remote server> element in the security platform XML configuration file.
5. If you are implementing security using an NT LAN Manager provider and you want to enable authentication of users from multiple Windows domains, but you do not want to set up trust relationships between those domains, install the Remote Authentication Module on a separate Windows server. This enables users of OLAP applications running on one domain to log in to OLAP applications on other domains. All the domains involved must be running applications that are configured to use the same Remote Authentication Module instance.
6. With the exception of the Remote Authentication Module, all security platform components are installed by default with DB2 OLAP Server. These include:

- `css-2_5_x.dll`, a library for enabling NT LAN Manager support. This file is installed to `HYPERION_HOME\common\CSS\2.5.x\bin`.
 - `css-full.xml`, a sample configuration file that you can modify or re-create to represent your authentication needs. This file is installed to `HYPERION_HOME\common\CSS\2.5.x\configuration`.
 - `css-2_5_x.jar`, containing the security platform classes. This file is installed to `HYPERION_HOME\common\CSS\2.5.x\lib`.
7. Create a security platform XML configuration file, as described in the Security Platform Reference section of the *Analytic Services Technical Reference*, under *Configuring the Security Platform*. The `css-full.xml` file might be used as an example to edit, or you can copy and edit `sample.xml` from the *Analytic Services Technical Reference*.
 8. Edit the `essbase.cfg` configuration file to point to the location of the XML file mentioned in step 7. This procedure is described in the Security Platform Reference section of the *Analytic Services Technical Reference*, under *Selecting the Authentication Module*. If Java, the XML file, and the `essbase.cfg` file are all configured correctly, then Analytic Server displays the following message upon startup:


```
Single Sign-On Initialization Succeeded !
```
 9. If you have installed Administration Services, edit the Administration Services file `OlapAdmin.properties` to point to the location of the properly configured XML file mentioned in 7. `OlapAdmin.properties` is located in the `eas\server` directory of the Administration Services installation. For example:


```
SECURITY_CONFIGURATION=file://c:/ibm/db2olap/bin/sample.xml
```
 10. You can now create users whose type is external. These users can log in to DB2 OLAP Server using the user names and passwords already assigned to them in the corporate authentication repository. For information about creating external users with MaxL, see the Security Platform Reference section of the *Analytic Services Technical Reference*, under *MaxL Statements for External User Management*. For information about creating external users with Administration Services, see the Administration Services Online Help.

Chapter 4. Installing and starting on UNIX

This chapter describes how to install server components for DB2 OLAP Server and DB2 OLAP Integration Server Version 8.2 on AIX, Solaris Operating Environment, HP-UX, and Linux operating systems.

Ensure that you consider the following issues about how DB2 OLAP Server is installed in version 8.2.

Upgrading to version 8.2 on the same UNIX computer

If you have V8.1 of DB2 OLAP Server on your UNIX computer, and you want to install version 8.2 on that computer, it is recommended that you make a copy of your existing OLAP directory.

You can install version 8.2 on top of V8.1, and your existing OLAP applications will be available to the new or upgraded version. But before installing a new version to the same directory, IBM recommends that you back up the following files:

- Your data files
- Essbase.sec
- license.*

Then you can install version 8.2 and import your application files and data into the new version.

Make sure you aware of all migration issues before starting the installation.

See Chapter 11, "Upgrading and migrating from earlier versions," on page 109.

This chapters contains the following topics:

- "Installing DB2 OLAP Server on UNIX"
- "Installing DB2 OLAP Integration Server on UNIX" on page 35
- "Installing DB2 OLAP Server Miner on AIX" on page 36
- "Configuring the environment on UNIX" on page 36
- "Starting the Analytic Server on UNIX" on page 39
- "Starting Integration Server on UNIX" on page 41
- "Troubleshooting common connection problems" on page 42
- "Adding additional software or users later" on page 43
- "Using agent commands" on page 43

Installing DB2 OLAP Server on UNIX

The user under which you are installing DB2 OLAP Server must have both write and execute privileges in the directory where you are running the installation. If you install the DB2 OLAP Server as root, then other users might have difficulty accessing the server and updating the installation with IBM Fix Packs, so it is recommended that you install under a user name other than root.

Ensure that you have adequate disk space available on your system. Run `df -k` to check the percentage of space available on the mount where you are installing DB2 OLAP Server.

During the installation, you will be asked to enter the DB2 OLAP Server license edition and options that you purchased. This information is used to build a license key for DB2 OLAP Server and to enable the products for which you have licenses.

Pre-installation steps

1. Log on to the server with a user ID that will be the administrator of DB2 OLAP Server. Do not use root.
2. If you are installing on a Solaris operating system, set the locale to c. When you have finished installing, make sure the locale is still c when you start the server for the first time. Using the correct locale ensures that the administration user ID is configured correctly.
3. If your workstation has a CD-ROM drive, follow these steps:
 - a. Insert the DB2 OLAP Server CD into your CD-ROM drive.
 - b. Use the `ls` command to find `setup.sh` on the CD. The file name will be either uppercase or lowercase, depending on your operating system.
 - c. Enter `setup.sh` to run the setup script from the DB2 OLAP Server CD.
4. If your workstation does not have a CD-ROM drive, you must create a temporary directory for use during the installation. To create a temporary directory, follow these steps:
 - a. Use FTP, NFS, or any similar utility to copy the DB2 OLAP Server installation files into your temporary directory.
 - b. Change to the temporary directory.
 - c. Use the `ls` command to find `setup.sh` in the temporary operating system subdirectory. For example, look in `/temp/aix` for the AIX installation file. The file name will be either uppercase or lowercase, depending on your operating system.
 - d. Enter `setup.sh` to run the setup script from the temporary directory.
5. If your operating system is not compatible with DB2 OLAP Server, you will receive a notification from the installation, which will then exit. If a multi-threaded environment is not detected, you will receive an error message.

Installation steps for DB2 OLAP Server

To install DB2 OLAP Server:

1. If you have never installed V8.2 OLAP products, the installation program displays the following prompt:

Enter the full path of the directory to place
the third party components (HYPERION_HOME folder):

Your answer sets the `HYPERION_HOME` environment variable and defines the directory in which to store some of the product files. Additionally, it stores files that are common to DB2 OLAP Server and its features in the `$HYPERION_HOME/commondirectory`.

If you have previously installed another V8.2 OLAP product, the `HYPERION_HOME` environment variable is already set. The installation displays a prompt similar to this one:

The Hyperion Home Directory (HYPERION_HOME folder) is: `/homes/db2olap`

2. Enter the directory where you want to install DB2 OLAP Server. This directory can be pre-existing, or a directory that will be created by the installation program. IBM strongly recommends that you install DB2 OLAP Server and its features into a new separate directory; do not install into directory in which a product other than DB2 OLAP Server and its features has been installed.

3. If you have a current DB2 OLAP Server license in another directory that you want to use for the current installation, enter that directory during installation for licensing reference.
4. In the License Agreement window, enter or select the following information:
 - a. Select the number that corresponds with the language in which you would like to read the IBM license.
 - b. Press **Enter** to view the license for approval.
 - c. Press **1** to accept the agreement, **2** to decline it, or **99** to return to the previous screen.

The IBM License Agreement Utility will then complete the licensing. The installation will continue to the next panel if the license was accepted, or exit the installation if the license was declined.

5. Select the license edition that you purchased.
6. Select the license options that you purchased, including the number of users licensed.
7. Select the products that you want to install.
8. After verifying that the summary is correct, select Continue.
9. Select Install to run the installation.

The installation directory includes a log file (olapinst.log) located in <ARBORPATH>/install, and a response file (olapisnt.rsp) that can be used for silent or unattended installations. The log file includes the license information and a list of the installed products. Additionally, it includes any installation errors if they occurred.

Installing DB2 OLAP Integration Server on UNIX

On UNIX, you can install DB2 OLAP Integration Server as an optional add-on when you install DB2 OLAP Server, or you can install it separately.

To install the DB2 OLAP Integration Server:

1. Verify that you have the necessary software product components and that the server and client computer systems on which you will install DB2 OLAP Integration Server meet system requirements. See Chapter 2, "Hardware and software requirements," on page 7.
2. Install DB2 OLAP Server if you have not already done so. DB2 OLAP Server does not have to be on the same computer as DB2 OLAP Integration Server.
3. Install the server component of DB2 OLAP Integration Server from the DB2 OLAP Server CD.
4. Install the client component (the Essbase Integration Services Console) from the DB2 OLAP Server CD, on a Windows computer.
5. Configure a data source using ODBC. See Chapter 5, "Configuring DB2 OLAP Server for relational integration," on page 45. On the computer running DB2 OLAP Integration Server, create Open Database Connectivity (ODBC) connections to an OLAP Metadata Catalog data source and to the relational data source that you use for creating OLAP models and metaoutlines.
6. Create an OLAP Metadata catalog. See Chapter 9, "Creating, upgrading, and deleting OLAP Metadata Catalogs," on page 79 for more information.
7. Set up the sample applications. See Chapter 10, "Setting up the sample application for DB2 OLAP Integration Server," on page 87.

Installing DB2 OLAP Server Miner on AIX

DB2 OLAP Server V8.2 includes DB2 OLAP Server Miner 8.1.2, also called OLAP Miner, which is a no-charge feature of DB2 OLAP Server that brings together IBM's data mining and OLAP technologies. OLAP Miner applies a sophisticated data mining algorithm called Deviation Detection to the data stored in OLAP cubes. The algorithm scans slices of cubes, identifying values that deviate from the norm. The results of a data mining run can be used by the business analyst to focus on the interesting or surprising values in context of the source OLAP cube. You can still use the usual slice, dice, and drill operations to explore the data, but now you let the values discovered by OLAP Miner drive your analysis. See the *OLAP Miner User's Guide* for information about what OLAP Miner can do for you.

OLAP Miner requires the DB2 OLAP Server API to be installed on the computer on which you install the OLAP Miner server. Java is a requirement for both DB2 OLAP Server and OLAP Miner.

OLAP Miner has a client component that runs on Windows and a server component that runs on both Windows and AIX. For information about installing the OLAP Miner client component on Windows, see "Installing DB2 OLAP Server Miner on Windows" on page 22. The OLAP Miner server is not supported on Solaris Operating System, HP-UX, or Linux.

DB2 OLAP Server Miner does not support migration from V8.1 to V8.1.2. If you previously installed V8.1 and want to upgrade to V8.1.2, you must first manually delete the V8.1. product.

To install the server component of OLAP Miner on AIX:

1. Insert the DB2 OLAP Server installation CD into your CD-ROM drive.
2. Find the `OLAP_Miner\aix` directory on the CD.
3. Run the `omsetup.sh` file to begin the installation program.
4. The installation program prompts you for the following information:
 - The alias name of the OLAP server component of DB2 OLAP Server.
 - The host address (or IP address) of the OLAP server component. The address is added to the `bin\omserver.cfg` file.

The installation directory includes a log file (`OMinstall.log`) located in `<ARBORPATH>/install`, and a response file (`minerisnt.rsp`) that can be used for silent or unattended installations. The `OMinstall.ini` file contains a recording of each installation and version of OLAP Miner.

Configuring the environment on UNIX

After installing DB2 OLAP Server and DB2 OLAP Integration Server on Linux, AIX, Solaris, or HP-UX, you must update the environment variables on your system. The installation program creates scripts you can use to apply the environment variables, and stores them in the `$ARBORPATH` or `$ISHOME` directory. The scripts are listed in Table 17 on page 37.

Table 17. Scripts for setting environment variables

Product	Shell	Script	Sets environment variables for:
DB2 OLAP Server	Korn or Bourne	essbaseenv.sh	DB2 OLAP Server; ODBC
	CShell	essbaseenv.csh	DB2 OLAP Server; ODBC
	Korn or Bourne	essjava.sh	Java
	CShell	essjava.csh	Java
DB2 OLAP Integration Server	Korn or Bourne	hisjava.sh	DB2 OLAP Server; ODBC; Java
	CShell	hisjava.csh	DB2 OLAP Server; ODBC; Java

If you run both DB2 OLAP Server and DB2 OLAP Integration Server on the same UNIX system, IBM recommends that you set the environment variables for each server component separately, in separate UNIX sessions. For example, you can take the following steps:

1. In the session running DB2 OLAP Server, add the `essbaseenv.sh` and `essjava.sh` scripts files to your `.cshrc` or `.profile` environment files.
2. In the session running DB2 OLAP Integration Server, add only the `hisjava.sh` file to your `.cshrc` or `.profile` file.

For example, in the `.profile` file, you can add the following command for each script:

```
. /full path/script name
```

where *full path* is the path where the script file is stored, and *script name* is the name of the script for the UNIX shell you use. After entering the statement, you must log out and then log back in to enable the new settings to take effect.

Alternatively, instead of updating the `.cshrc` or `.profile` files, you can manually invoke the `essbaseenv.sh` and `essjava.sh` scripts each time you start a session to run DB2 OLAP Server, and in the session running DB2 OLAP Integration Server, invoke only the `hisjava.sh` script.

The `$ARBORPATH/essbaseenv.doc` file contains additional information about setting environment variables in your environment.

Optionally, you might need to take these additional steps:

- If you are licensed for SQL Interface or Hybrid Analysis and plan to run either on any UNIX platform, you must take additional steps to set the environment. See Chapter 5, “Configuring DB2 OLAP Server for relational integration,” on page 45.
- If you use ODBC drivers other than those from DataDirect, you need to run a shell script that sets the library path to access the ODBC driver. See “Updating the DB2 OLAP Integration Server environment for ODBC” on page 38.
- If you are installing on Solaris, and you plan to use the Cache Memory Locking database setting, you must also run `root.sh` from the `ARBORPATH/bin` directory. See “Enabling use of cache memory locking systems” on page 38.
- If you want to perform scheduled member and data loads on UNIX, you must be authorized to use the `cron` scheduling daemon. See “Configuring the `cron` scheduling daemon” on page 39.

Updating the DB2 OLAP Integration Server environment for ODBC

The `hisjava.csh` and `hisjava.sh` scripts provided with DB2 OLAP Integration Server set the library path to access the DataDirect ODBC drivers provided with the product. If you use these drivers, include either `hisjava.csh` or `hisjava.sh` in the login script for the user who runs DB2 OLAP Integration Server to set the required variables.

If you use ODBC drivers other than DataDirect, the provider typically provides a shell script to set the library path to access the driver. Add this shell script to the login script for the user who runs the DB2 OLAP Integration Server software.

An example of a shell script that sets the library path to access ODBC libraries on Solaris follows:

```
if env | grep LD_LIBRARY_PATH" >
    $ISHOME/esslib:$HYPERION_HOME/common
    /ODBC/Merant/4.2/lib:$LD_LIBRARY_PATH
else
    LD_LIBRARY_PATH=
    $ISHOME/esslib:$HYPERION_HOME/ODBC/Merant/4.2/lib
```

Updating the environment for the database client

For DB2 OLAP Integration Server, set the environment variables required for database client software in the login script of the user who runs DB2 OLAP Integration Server. These environment settings are required for ODBC access to the databases in the RDBMS that you are using. The database vendor typically supplies a shell script to set any environment variables required by the database client. Add this shell script to the login script for the user who runs the DB2 OLAP Integration Server software. For more information about how to set environment variables for the database client, see the database client installation documentation.

To verify that the database client software is set up correctly, log on as the user who runs DB2 OLAP Integration Server, and use a database utility to connect to the databases that you use with DB2 OLAP Integration Server.

Enabling use of cache memory locking systems

If you are running Analytic Server on a Solaris system and you plan to use the Cache Memory Locking database setting, you must run the Bourne shell script `root.sh`.

It is not necessary to run `root.sh` unless you plan to use the Cache Memory Locking database setting on a Solaris system.

CAUTION: Using `root.sh` may interfere with SQL Interface. You may need to choose between SQL Interface and cache memory locking.

After you install Analytic Server, run `root.sh` from the root account. The script, located in `ARBORPATH/bin`, performs post-installation tasks. To run `root.sh`, log on as user root, and type the following commands:

```
$ su -
Password:
# cd /home/hyperion/essbase/bin
# sh ./root.sh
# exit
```

The script directs the following actions:

- Runs as root so that it has appropriate access to files and directories.
- Changes ownership of ESSVR, the application server process, to root. This action is necessary to enable use of the optional Cache Memory Locking setting.
- Creates symbolic links in /usr/lib to the following shared libraries in ARBORPATH/bin:

On Solaris:

```
libessd.so.1  
libessapi.so.1  
libessnetu.so.1  
libessotl.so.1  
libessshr.so.1  
libesssql.so.1  
libessutl.so.1  
libesssvrd.so.1  
libesssm.so.1  
libessdvrq.so.1  
libglobalc.so
```

:

Configuring the cron scheduling daemon

DB2 OLAP Integration Server uses the cron scheduling daemon to perform scheduled member and data loads. This daemon must be started on the computer that runs DB2 OLAP Integration Server and the user who starts the program must be authorized to schedule cron jobs. For more information, see the documentation for the UNIX system you are using.

In order to schedule jobs for DB2 OLAP Integration Server, the crontab file must exist. You can check whether this file exists by running the following command in your environment:

```
crontab -l
```

If this file does not exist in your environment, you can create it manually by using the following command:

```
crontab -e
```

At this point, the file is open for editing. You can quit the file without saving it. For details on crontab settings, refer to the help for the UNIX operating system you are using.

Starting the Analytic Server on UNIX

You can start the server in the foreground or in the background.

Before you can access the server, you must be logged on to the UNIX server on an account that has at least Read/Write/Execute access to the Essbase directories. Ideally, this is the same account that was used to install DB2 OLAP Server.

After you start Analytic Server, all Analytic Server activities are recorded in the Analytic Server log (essbase.log), which is stored in the ARBORPATH directory. After you start individual applications, application and database activities are recorded in an application log (appname.log), which is stored in the application directory (\$ARBORPATH/app/appname).

If you have problems connecting to Java on HP-UX, or if DB2 OLAP Server ends abnormally on HP-UX, see “Troubleshooting common connection problems” on page 42.

Starting Analytic Server in the foreground

To start the Analytic Server in the foreground:

1. From the command prompt, type:

```
$ARBORPATH/bin/ESSBASE
```

or, if your path includes \$ARBORPATH/bin, type:

```
ESSBASE
```

If you encounter any errors during startup, check your path and environment settings.

2. If this is a first-time installation, Essbase prompts you for the following items:
 - **Company name.** The name that you type will be embedded in your server license registration.
 - **Your name.** The name that you type will be designated as the Essbase system supervisor. Use this name for your initial logon from the client to the server.
 - **System password.** The password that you choose is required every time you start the Essbase server.
 - You are asked to confirm your choices. Type **1** to verify your entries, or **2** to retype them.
3. If this is not a first-time installation, the system prompts you for the system password that you specified when you first ran the server.

The server then displays the following message:

```
Waiting for Client Requests...
```

You can now use the Essbase Agent to perform high-level supervisor operations.

Starting the server in the background

If you are starting the Analytic Server for the first time after installing, you must start it in the foreground once before you can start it in the background. You must start Analytic Server in the foreground the first time after installing because you need to be prompted for a company name, an administrator user name, and a password. Analytic Server requires this startup information to run. However, if you installed Version 8.2 in the same directory as an existing installation, it is not necessary to start the server in the foreground before starting it in the background.

After you have started the server in the foreground and followed the prompts to enter your company name, Essbase administrator user name, and password, type EXIT to stop the server and follow the instructions to start the server as a background process.

To start the server as a background process:

1. If this is a first time installation, you must start the server in the foreground.
2. To start the Analytic Server in the background on UNIX, or on Windows systems utilizing a UNIX-like shell such as MKS, enter the following command at the command prompt of your operating system:

```
$ARBORPATH/bin/ESSBASE password -b &
```

If your path already includes \$ARBORPATH/bin, type this instead:

```
ESSBASE password -b &
```

You can start the Analytic Server without using the ampersand (&) at the end of the command, but if you do not type the &, the command prompt is not returned after the server is started.

Hiding Analytic Server passwords on UNIX

To hide the system password from the UNIX process listing created by the `ps -ef` utility on Solaris and HP-UX, you must create a simple shell script and use it to launch the Analytic Server agent. Create a shell script named `essbase.secure`, consisting of the following lines:

```
#!/bin/sh
PASS=$1
ESSBASE -b -secure << EOF &
${PASS}
EOF
```

After creating this script, launch the Agent to start the script in the background. The script obtains the password from process listing details:

```
essbase.secure password
```

Note: On AIX servers, there is no need to hide the Essbase system password because it is hidden automatically.

Shutting down Analytic Server

To shut down the server when Analytic Server is running in the foreground, type `EXIT`.

To shut down the server from any terminal connected to it, use the `SHUTDOWNSERVER` command in `ESSCMD`.

Starting Integration Server on UNIX

After you install the server software, you can run Integration Server or Integration Services Shell from any directory.

You can start Integration Server and automatically configure the environment, or you can manually configure the environment before starting Integration Server.

Running Integration Server on UNIX Systems

To run Integration Server in the foreground and automatically update your environment, type the following command:

```
olapisvr
```

To run Integration Server in the background, modify `olapisvr` or type the following commands:

```
nohup olapisvr &
hisjava.sh
```

The `nohup` command keeps the Integration Server program running even if you log off. The ampersand (&) suffix makes the program a background process.

For more information on starting and stopping Integration Server from the command line, on other startup switches for the `olapisvr` command, and on using the `eis.cfg` file, see the *Integration Services System Administrator's Guide*.

CAUTION: To allow the Integration Services Console client to create OLAP models and metaoutlines, you must run Integration Server and keep it running while any clients are connected to it.

When you start Integration Server, it begins writing to the log, `olapisvr.log`, in the log directory. To write to a log on another directory, type the following command:

```
olapisvr -Emdir/mylog
```

Do not type the `.log` file extension. The `.log` extension is automatically appended to the name that you type for the log.

Stopping Integration Server on UNIX Systems

You can use the SHUTDOWN command in the Integration Services Shell (`olapicmd`) to stop Integration Server. For more information, see the *Integration Services System Administrator's Guide*.

Troubleshooting common connection problems

If you have connection problems after completing the installation, check the following list for possible causes and solutions:

TCP/IP

Before attempting a client-server connection, test the protocol installation to ensure that the connection is installed properly.

The servers

Use the PING command on the server to check that it is running and visible to the client. If the PING command is successful but the TELNET command is not, then there may be a problem with the `inet` daemon. If the PING command is unsuccessful on the server, you might have a routing or hardware problem.

General problems when connecting to Java

After following the instructions in "Configuring the environment on UNIX" on page 36, if you are still having difficulties getting DB2 OLAP Server to work with Java, it may be necessary to set startup options for the Java Virtual Machine (JVM). You can make DB2 OLAP Server use specific JVM startup options when it loads the JVM by setting the environment variables `ESS_JVM_OPTION1` through `ESS_JVM_OPTION9`.

Problems with Java on Solaris 1.4.1_06 only

The `ESS_JVM_OPTION1` variable is required on Solaris 1.4.1_06 if you are using JVM for custom-defined functions, custom-defined macros, data mining, triggers, or external authentication supplied by Hyperion. You must set an `ESS_JVM_OPTION` environment variable to use a special Java argument, `Xusealtsigs`. This requirement applies only to JRE version 1.4.1_06 on Solaris and is not applicable to other JRE versions on Solaris. To set `ESS_JVM_OPTION1`:

- In the C shell, add the following text to the `.cshrc` file:

```
setenv ESS_JVM_OPTION1 "-Xusealtsigs"
```
- In the Bourne or Korn shell, add the following text to the `.profile` file:

```
ESS_JVM_OPTION1=-Xusealtsigs; export ESS_JVM_OPTION1;
```

Additionally, `ESS_CSS_JVM_OPTION1` is required on Solaris 1.4.1_06 if you are using the security platform supplied by DB2 OLAP Server. You must set an `ESS_CSS_JVM_OPTION` environment variable to use a special Java argument, `Xusealtsigs`. To set `ESS_CSS_JVM_OPTION1`:

- In the C shell, add the following text to the `.cshrc` file:

```
setenv ESS_CSS_JVM_OPTION1 "-Xusealtsigs"
```
- In the Bourne or Korn shell, add the following text to the `.profile` file:

```
ESS_JVM_CSS_OPTION1=-Xusealtsigs; export ESS_CSS_JVM_OPTION1;
```

Problems with Java on HP-UX

If you have problems connecting to Java on HP-UX, or if DB2 OLAP Server ends abnormally, set the environment variable `LD_PRELOAD`. Set this variable in a shell script that you design for starting Analytic Services. For example, create a script called `startessbase.sh` in `$ARBORPATH/bin` with the following contents (for Bourne or Korn shell):

```
export LD_PRELOAD=HYPERION_HOME/common/JRE/1.4.1/lib/  
PA_RISC2.0/server/libjvm.sl  
$ARBORPATH/bin/ESSBASE
```

Then start DB2 OLAP Server using the startup script:

```
cd $ARBORPATH/bin ./startessbase.sh
```

The following example of a `startessbase.sh` script is for the C shell:

```
setenv LD_PRELOAD LD_PRELOAD$/home/hyperion/common/JRE/  
1.4.1/lib/PA_RISC2.0/server/libjvm.sl  
$ARBORPATH/bin/ESSBASE
```

CAUTION:

Setting `LD_PRELOAD` in UNIX environment scripts such as `.profile` is not recommended, because side effects may occur.

Adding additional software or users later

The DB2 OLAP Server installation program prompts you for the edition and options that you have purchased, and uses that information to enable DB2 OLAP Server and add-on components.

If you install a license of DB2 OLAP Server and later want to add more features or additional users, run the `license` program from the `/bin` directory of the DB2 OLAP Server installation directory. The `license` program will automatically update the license.

Using agent commands

DB2 OLAP Server has a server process called the Agent. When you start the main server program (`ESSBASE`), the Agent becomes the active terminal, unless you start the server in the background. The Agent acts as a console for high-level server operations. Pressing the Enter key in the Agent terminal provides a list of all available commands.

Note: The Agent commands are available only when the server is started in the foreground.

The following table shows the commands that are used to manage DB2 OLAP Server.

When you start DB2 OLAP Server, the Agent window opens. The Agent window is the console for high-level server operations. Press Enter to view the following list of all available commands.

Table 18. Agent Commands

Command	Description
START <i>appname</i>	Starts the specified application.
STOP <i>appname</i>	Stops the specified application.
USERS	Displays a list of all users that are connected to the server, the total number of connections, and the number of ports that are available.
LOGOUTUSER <i>username</i>	Disconnects a user from the server and frees a port. This command requires the DB2 OLAP Server password.
PASSWORD	Changes the system password that is required to start DB2 OLAP Server.
VERSION	Displays the server version number.
HELP	Lists all valid commands and their respective functions.
PORTS	Displays the number of ports that are installed on the OLAP server and how many are in use. Number of available ports = number of licensed ports + 1 system administrator port.
DUMP <i>filename</i>	Dumps information from the DB2 OLAP Server security system to a specified file in ASCII format. If you do not supply a path with the file name, the file is saved to the /\$ARBORPATH/bin directory. This command requires the DB2 OLAP Server password.
QUIT/EXIT	Shuts down all open applications and stops DB2 OLAP Server.

Chapter 5. Configuring DB2 OLAP Server for relational integration

Hybrid Analysis and SQL Interface are tools you can use to integrate relational databases and OLAP databases, thereby leveraging the scalability of the relational database with the conceptual power of the multidimensional database. This chapter describes the tasks required to configure the computer on which you run DB2 OLAP Server to enable SQL Interface and Hybrid Analysis.

Although SQL Interface and Hybrid Analysis are two separate products with different functionality, many of the tasks required to configure the computer on which they are used are the same. Those tasks are covered in this chapter, as well as some separate tasks that are required for each product. Additionally, this chapter includes a Feature Map, which is a table that summarizes all the tasks associated with Hybrid Analysis.

This chapter contains the following topics:

- “Preparing to use SQL Interface or Hybrid Analysis”
- “SQL Interface and Hybrid Analysis on Windows”
- “SQL Interface and Hybrid Analysis on UNIX” on page 46
- “Additional tasks” on page 48
- “Hybrid Analysis features map” on page 49

Preparing to use SQL Interface or Hybrid Analysis

Before configuring DB2 OLAP Server to run Hybrid Analysis or SQL Interface, consider the following information:

- On Windows, each DataDirect ODBC driver has its own help file. You can access files using the ODBC Administrator. See the *Analytic Services SQL Interface Guide*.
- The DataDirect ODBC drivers come with a document called DataDirect Connect ODBC Reference. This reference is in PDF for online viewing and for printing in Adobe Acrobat Reader (Version 3.0.1 or later).
- You may also need to configure your data source. For detailed information on configuring data sources for DB2 OLAP Server, see the *Analytic Services SQL Interface Guide*. For detailed information on configuring data sources for DB2 OLAP Integration Server, see Chapter 8, “Configuring relational data sources for DB2 OLAP Integration Server,” on page 63.
- You must meet the specific system requirements for SQL Interface and Hybrid Analysis. See “Supported ODBC drivers and relational data sources” on page 14 for detailed information about system requirements for each.

Some ODBC drivers and relational data sources that are supported for SQL Interface are not supported for Hybrid Analysis.

SQL Interface and Hybrid Analysis on Windows

The installation program places all SQL Interface program files in the ARBORPATH\bin directory on. This includes `esssql.dll`.

ODBC driver files are needed for both SQL Interface and Hybrid Analysis. The ODBC driver files and their corresponding help (.hlp) files are installed in the HYPERION_HOME directory.

If the data source is Teradata, you must also make sure that the Teradata environment is set up correctly. You must be able to connect to Teradata from the Analytic Server computer before SQL Interface or Hybrid Analysis can work. Confirm that the Teradata server is installed and running, and set the environment variables required to run Teradata.

SQL Interface and Hybrid Analysis on UNIX

The following list is an overview of tasks required to run SQL Interface or Hybrid Analysis on UNIX:

1. Confirm that you meet system requirements. See “Supported ODBC drivers and relational data sources” on page 14.
2. Install the Analytic Server.
3. From the \$ARBORPATH directory, run **inst-sql.sh** to link SQL Interface to the ODBC driver. See “Linking SQL Interface to ODBC drivers on UNIX” on page 47 for detailed information and instructions.
4. If using Hybrid Analysis, set the \$ODBCINI environment variable to the path and file name of the ODBC data source configuration file.
On the AIX, Solaris, and HP-UX operating systems, SQL Interface always expects to find the file in \$HOME/.odbc.ini and will ignore the value of \$ODBCINI. If you are using SQL Interface on Linux, or if you are using both Hybrid Analysis and SQL Interface on any UNIX platform, save the file as \$HOME/.odbc.ini and set \$ODBCINI to \$HOME/.odbc.ini.
5. Edit \$ARBORPATH/bin/.odbc.ini to include the correct driver and data source names.
6. If Teradata is the data source, see “Setting up Teradata on UNIX” on page 47 for additional steps required before running SQL Interface.

What happens during installation on UNIX

The installation program installs the SQL Interface files and additional files, such as ODBC drivers, required to run both SQL Interface and Hybrid Analysis.

The installation program installs DataDirect ODBC drivers files to \$HYPERION_HOME/ODBC/Merant/4.2.

- The following SQL Interface files are stored in the \$ARBORPATH/bin directory:
 - libesssql.insolv.a on AIX
 - libesssql.insolv.sl on HP-UX
 - libesssql.so.1 on Solaris
 - libesssql.insolv.so on Linux

When the Analytic Server is installed on UNIX, it creates:

- The SQL Interface library file:
 - libesssql.a on AIX
 - libesssql.sl on HP-UX
 - libesssql.so on Solaris
 - libesssql.so on Linux
- A stub ODBC drivers file:

- libesssql.arbor.a on AIX
- libesssql.arbor.sl on HP-UX
- libesssql.arbor.so.1 on Solaris
- libesssql.arbor.so.1 on Linux

After you install Analytic Server, you must run the shell script `inst-sql.sh` to enable SQL Interface. The script creates a symbolic link between the SQL Interface library file and the Data Direct ODBC drivers file.

Linking SQL Interface to ODBC drivers on UNIX

After you install Analytic Server and set up the Analytic Server environment variables, but before running SQL Interface, you must link SQL Interface to the ODBC drivers by running the shell script `inst-sql.sh`. Running `inst-sql.sh` ensures that server processes can access the software necessary for SQL Interface.

The `inst-sql.sh` script uses the `odbc.ini`, `odbcinst.ini`, and template files in the `$HYPERION_HOME/common/ODBC/Merant/4.2/lib` directory to create a set of files that points Analytic Services to the correct driver product.

The script performs the following tasks for you:

- Ensures that root privileges are not accidentally granted to a user-level script.
- Checks for the availability of Analytic Server.
- Checks to see if supported database systems are available to Analytic Server.
- Provides initialization files in the `$ARBORPATH/bin` directory for access by the driver software.
- Creates links for the following files:
 - libesssql.insolv.a on AIX
 - libesssql.insolv.sl on HP-UX
 - libesssql.so.1 on Solaris
 - libesssql.insolv.so on Linux
- Copies the file `$HYPERION_HOME/ODBC/Merant/4.2/odbc.ini` into `$ARBORPATH/bin/.odbc.ini`.
- Links `.odbc.ini` in your home directory to `$ARBORPATH/bin/.odbc.ini`.

Setting up Teradata on UNIX

You must be able to connect to Teradata from the Analytic Server computer before SQL Interface or Hybrid Analysis can work.

Confirm that the following tasks have been performed:

1. The Teradata ODBC is installed on the computer where the Analytic Server is running.
2. The environment variable is updated to point to the Teradata ODBC. See “Supported ODBC drivers and relational data sources” on page 14 for information on tested and supported SQL Interface platforms and ODBC drivers.
3. Set the environment required to run Teradata.

Completing configuration tasks on UNIX

After installing DB2 OLAP Server and running `inst-sql.sh` (if you are planning to use SQL Interface), you are ready to complete the configuration tasks. These tasks are required for both SQL Interface and Hybrid Analysis.

- Set the library path. See “Setting library paths on UNIX.”
- On Linux, set the ODBCINI environment variable.
- Complete the configuration of the ODBC drivers by editing the `.odbc.ini` file. See “Editing ODBC configuration files” for examples.

Setting library paths on UNIX

To run SQL Interface or Hybrid Analysis on UNIX, you must add the `$HYPERION_HOME/common/ODBC/Merant/4.2/lib` directory to the library path. If the installation of DB2 OLAP Server is an upgrade to a pre-existing installation, you may not have to reset the library path.

Setting the ODBCINI environment variable on Linux

Unlike other supported UNIX platforms, on Linux, you must set the ODBCINI environment variable to use SQL Interface.

To use Hybrid Analysis, you must set the ODBCINI environment variable on all supported UNIX platforms.

Editing ODBC configuration files

To complete the ODBC driver setup, you must modify `$ODBCINI` to include the correct driver and data source names. This is required for both SQL Interface or Hybrid Analysis. See the following examples for the IBM DB2 driver on AIX and for the Data Direct DB2 driver on Solaris.

Editing the ODBC configuration file for Data Direct Oracle drivers on Solaris

To set up the `.odbc.ini` file for Oracle 8 on Solaris, modify `$ODBCINI` to include the correct driver and data source name. For example:

```
[ODBC Data Sources]
Oracle8=MERANT 4.20 Oracle 8 Driver

[Oracle8]
Driver=$HYPERION_HOME/common/ODBC/Merant/4.2/lib/ARor819.so
Description=Oracle8
ServerName=name
```

Where *name* is the appropriate Oracle server name.

Additional tasks

After configuring the Analytic Server computer to use SQL Interface or Hybrid Analysis, you must complete some additional tasks before beginning to use them. The following topics contain a summary of the required tasks.

Additional Hybrid Analysis tasks

Before using Hybrid Analysis:

- In the `essbase.cfg` file, set `HAENABLE TRUE`. See the `HAENABLE` topic in the *Analytic Services Technical Reference* for syntax and usage guidelines.
- In Spreadsheet Add-in options, on the **Zoom** tab, select the option to enable Hybrid Analysis.

- Ensure that the name of the ODBC Data Sources used to connect to both the OLAP Catalog and the data source database are exactly the same on the Analytic Server as those used on the Integration Server to build the outline. The user names and passwords for connecting to those databases should also be identical.
- Build the Analytic Services database with a DB2 OLAP Integration Server metaoutline that invokes Hybrid Analysis. See the DB2 OLAP Integration Server documentation for detailed information on how to complete these tasks.

Additional SQL Interface tasks

Before using SQL Interface to load SQL, relational, or flat-file data, you must perform additional preparation tasks. See the *Analytic Services SQL Interface Guide* for detailed information and instructions for each of these tasks.

- Configure the ODBC driver and point it to its data source.
- Check that you can connect to the data source by using the Data Prep Editor in Administration Services Console to open the SQL source file.
- Create a data load rules file to tell SQL Interface how to interpret and load the SQL data into the Analytic Services database.

Hybrid Analysis features map

Table 19 lists the documents where you can find information about topics associated with Hybrid Analysis:

Table 19. Hybrid Analysis topics and where they are documented

Topic	Where it is documented
Latest updates on Hybrid Analysis: <ul style="list-style-type: none"> • New features • Enhancements • Issues 	<ul style="list-style-type: none"> • DB2 OLAP Server release notes (includes information about DB2 OLAP Integration Server) • Essbase Integration Services New Features • Essbase Analytic Services New Features
How to set up Hybrid Analysis: <ul style="list-style-type: none"> • Relational database support • Operating system support • Sample Application 	In this book: <ul style="list-style-type: none"> • This chapter. • Chapter 2, “Hardware and software requirements,” on page 7 • Chapter 6, “Setting up the sample application for DB2 OLAP Server,” on page 51
Fundamental concepts of Hybrid Analysis: <ul style="list-style-type: none"> • Overview and benefits • Relational sources • Data retrieval • The outline editor • Data consistency • Guidelines • Formulas • Supported functions 	<i>Analytic Services Database Administrator's Guide</i> , the section about <i>Accessing Relational Data with Hybrid Analysis</i>
How to perform Hybrid Analysis procedures	DB2 OLAP Integration Server online help

Table 19. Hybrid Analysis topics and where they are documented (continued)

Topic	Where it is documented
How to retrieve Hybrid Analysis data using these tools: <ul style="list-style-type: none"> • Spreadsheet Add-in for Excel • DB2 OLAP Server Spreadsheet Services 	<ul style="list-style-type: none"> • Spreadsheet Add-in online help • DB2 OLAP Server Spreadsheet Services online help
Configuration settings and MaxL grammar	<i>Analytic Services Technical Reference</i>

Chapter 6. Setting up the sample application for DB2 OLAP Server

This chapter describes how to set up the sample applications for DB2 OLAP Server.

DB2 OLAP Server offers the following sample applications. These sample applications are based on The Beverage Company (TBC), a fictional company that manufactures and sells soft drink products.

- The main application used in the documentation is called Sample. It contains three databases: Basic, Interntl, and Xchgrate. The latter two contain specific features corresponding to the currency conversion option.
- The Demo application contains one database, Basic, which is referred to in the *Analytic Services Technical Reference*.
- The Samppart application, is the first of two applications designed to show the features of Partitioning. Samppart contains a Company database that is the target of a partitioned database.
- The Sampeast application, is the second of two applications designed for Partitioning. Sampeast contains an East database that is the source of data for a partitioned database.
- The Sample_U application demonstrates a Unicode-mode version of Sample Basic, including alias tables in four different character sets, in addition to English. Characters in the Sample_U application are encoded in UTF-8.
- The DMDemo application demonstrates the features of Data Mining. DMDemo contains one database, called Basic. The sample mining tasks in the data mining documentation use data in Basic to illustrate data mining functionality.
- The ASOsamp application is designed to show the features of aggregate storage databases. ASOsamp contains one database, called Sample.

If you plan to use the sample applications, you must perform additional steps to prepare them for use in your organization. This chapter contains the following topics:

- “Loading the sample databases for DB2 OLAP Server”
- “Providing user access to the sample applications” on page 52
- “Preparing the sample aggregate storage application for use” on page 53
- “Setting the environment for the partitioning sample applications” on page 53
- “Loading data into Samppart and Sampeast” on page 55

Loading the sample databases for DB2 OLAP Server

To prepare the sample databases for use, you must load data into them. Each of the sample databases has a corresponding data load file.

Each of the sample databases has a corresponding data load file. Table 20 on page 52 shows the sample applications, the databases that are contained within them, and their corresponding data load files:

Table 20. Sample databases and data load files

Application	Database	Data load file
Sample	Basic	Calcdat
	Interntl	Currcalc
	Xchgrate	Rates
Demo	Basic	Data
Samppart	Company	Calccomp
Sampeast	East	Calceast
Sample_U	Basic	Calcdat.txt
DMDemo	Basic	Dmdemo.txt
ASOsamp	Simple	dataload.txt ¹

Notes:

1. The ASOsamp database must be loaded using the rules file, dataload.rul. The other samples do not require rules files to be loaded.

If you are loading data into the Partitioning sample databases, you must first create a partition user or other user name. See “Setting the environment for the partitioning sample applications” on page 53 for detailed information and instructions.

To load data into the sample databases, choose an application and database and the corresponding data load file from Table 20 and complete these steps.

1. Open the Administration Services Console.
2. From **Enterprise View**, find the appropriate Analytic Server and application.
3. From the **Databases** node, select the database.
4. Right-click and select **Load data**.
5. In the **Data Load** dialog box, select **Load data**.
6. In the **Data files** list box, click the **Find** button to locate a data file to load. The Open dialog box is displayed.
7. Select the **Analytic Server** tab.
8. Ensure that **List Files of Type** is set to Data files (*.txt).
9. Select a data load file and click **OK** to close the dialog box and load the data.
10. Click **OK** to load the data. The Data Load Completed dialog box is displayed.

Providing user access to the sample applications

DB2 OLAP Server provides a comprehensive security system for a secure multiple user environment. By default, the sample applications are created with a security access level of None, which means that no user can connect to the sample databases unless the user is defined as a supervisor.

The system supervisor, defined when the server component was installed, automatically has supervisor privilege. Therefore, the system supervisor can make the sample applications available to other users.

To provide all users with Write access to the sample databases, repeat the following procedure for each database.

1. Log on using the system supervisor account.

2. Open the Administration Services Console.
3. From **Enterprise View**, find the appropriate server and application.
4. Select the application, right click it, and select **Edit properties**.
5. In the Application Properties window, select the **General** tab.
6. For the **Minimum access level** option, select the **Write** option.
For example, if you want all users to have at least Write access to all databases in the application (meaning that all users can update data values), select **Write**.
7. From the **Minimum Database Access** group, select the **Write** option.
8. Click **Apply**.

The applications are now ready for use. If you want to provide access to another application, repeat the procedure. If you want to use the Partitioning applications, follow the instructions in “Setting the environment for the partitioning sample applications.” If you do not want to use the Partitioning applications, your next step might be to create the appropriate user names to be able to log on to DB2 OLAP Server. See the *Analytic Services Database Administrator’s Guide* for information on creating users.

Preparing the sample aggregate storage application for use

To prepare the sample aggregate storage application for use, you need to load data and then precalculate and store aggregations for the database.

To load data, use the following data file and rules file that are installed in the database directory on the Analytic Server:

```
ARBORPATH\app\AS0samp\Sample\data\load.txt  
ARBORPATH\app\AS0samp\Sample\data\load.rul
```

For instructions on loading data to aggregate storage databases, see Administration Services online help.

After the data load is complete, precalculate aggregations on the database to improve retrieval times. To precalculate aggregations, use Aggregation Design Wizard in Administration Services or use the MaxL execute aggregate process statement. For more information, see the Administration Services online help or the MaxL section of the *Analytic Services Technical Reference*.

Setting the environment for the partitioning sample applications

DB2 OLAP Server includes two sample applications and databases that show the features of Partitioning:

- Samppart, Company
- Sampeast, East

These applications and databases include partition definitions stored in .ddb files. The .ddb files define the map between member combinations in the target database, Company, and the source database, East. The *Analytic Services Database Administrator’s Guide* describes Partitioning, member combinations, and the differences between target and source databases.

For the Partitioning applications to work in your environment, you may need to create a user named `partitionuser`, or you may need to change the embedded user names in the .ddb files.

Note: Do not go into the .ddb files to change user name information. Instead, complete the steps in the following topics: “Creating the partition user” describes how to create the required user name, so that users can use the applications. “Changing embedded server names in the sample partition definitions” describes how to change the embedded server name to your server name.

Creating the partition user

You must create a user named Partitionuser before you work with the Samppart and Sampeast applications. Partitionuser must have access to both applications.

To create the Partitionuser:

1. Open Administration Services Console.
2. In the **Enterprise View**, select the appropriate Administration Server name under the **Administration Servers** node.
3. Under that node, select the **User** node, right-click, and select **Create user**.
4. In the **Create user on Administration Server** dialog box, in the **User** name text box, enter partitionuser.
5. In the **Password** text box, type password.
6. In the **Confirm Password** text box, type the password again.
7. In the **Administrator privileges** drop-down list box, select **true**.
8. Click **OK**.

Changing embedded server names in the sample partition definitions

If you choose not to create Partitionuser, you can change the embedded user name in the partition definition files (.ddb) to the user name of your choice as long as that user has administrator privileges. The .ddb files shipped with Samppart and Sampeast are based on the server name localhost.

To change the server name in the Samppart Company and Sampeast .ddb file:

1. Open Administration Services Console.
2. From the **Enterprise View**, expand the **Applications** node and select the **Samppart** application.
3. Expand the **Databases** node and select the **Company** database.
4. Expand the **Partitions** node and select **Source Databases**.
5. Under the Source Databases node, double-click:
`servername:SampEast:East [transparent]`
The Repair Partition dialog box opens.
6. In the **Data Source** group and the **Data Target** group, from the **User** drop down menu, select the user name of your choice.
7. Enter the password for the user.
8. Click **Repair** to save your changes.

Note: Note: If, in the Repair Partition dialog box, you changed the user name for both the **Data Source Group** and the **Data Target** group, you do not need to repeat this process to change the user name in the Sampeast East .ddb file.

Loading data into Samppart and Sampeast

After setting up the environment for the Samppart and Sampeast databases, you need to load data into each database.

1. Open Administration Services Console.
2. From **Enterprise View** or a custom view, select the database (the Samppart Company database or the Sampeast East database).
3. Right-click and, from the pop-up menu, select **Load Data**.
4. In the **Data Load** dialog box, click **Find** to browse for the correct data source. Use `Calccomp.txt` for Samppart Company, and use `Calceast.txt` for Sampeast East.
5. Select the data source and click **OK**.
6. In the **Data Load** dialog box, click **OK** to load the data.

Note: No rules files are required to load `Calccomp.txt` into Samppart Company or to load `Calceast.txt` into Sampeast East.

Chapter 7. Connecting DB2 OLAP Integration Server

This chapter describes how to start Integration Services Console and Integration Services Shell, and connect to DB2 OLAP Integration Server, DB2 OLAP Server, and external data sources.

This chapter contains the following topics:

- “Starting Integration Services Console”
- “Connecting to server components and data sources”
- “Connecting to Integration Server” on page 58
- “Connecting to Analytic Server” on page 59
- “Setting connection defaults” on page 60
- “Connecting to data sources” on page 60
- “Reconnecting when connections are lost” on page 62

To complete the installation of DB2 OLAP Integration Server, see Chapter 8, “Configuring relational data sources for DB2 OLAP Integration Server,” on page 63, and Chapter 9, “Creating, upgrading, and deleting OLAP Metadata Catalogs,” on page 79. To work with the OIS sample application, see Chapter 10, “Setting up the sample application for DB2 OLAP Integration Server,” on page 87.

Starting Integration Services Console

After you install the client software, complete the following procedure to start Integration Services Console.

To start Integration Services Console:

1. On the Windows desktop, click **Start**.
2. Select **Programs > IBM DB2 OLAP Server 8.2 > IBM DB2 OLAP Integration Server > Desktop**.

To display the Integration Services Console in a language other than English, change the Regional Options in Windows to the language you want.

Connecting to server components and data sources

Whether you are installing DB2 OLAP Integration Server for the first time or upgrading from a previous release, after you start the console, the OLAP Metadata Catalog Setup dialog box is displayed. Using this dialog box, you can either create a catalog or upgrade an existing catalog.

An OLAP Metadata Catalog must be created and configured before you can connect to it:

- For information on configuring a relational data source for an OLAP Metadata Catalog, see Chapter 8, “Configuring relational data sources for DB2 OLAP Integration Server,” on page 63.
- For information on creating OLAP Metadata Catalogs, see Chapter 9, “Creating, upgrading, and deleting OLAP Metadata Catalogs,” on page 79.

After the process of creating or upgrading an OLAP Metadata Catalog is complete, you can begin to create OLAP models and metaoutlines. You must connect to Integration Server to an external data source and to the OLAP Metadata Catalog where you want to store the OLAP models and metaoutlines that you create.

To use Integration Services Console to perform data loads, member loads, or member and data loads, you must connect the client software to the server component of DB2 OLAP Server, which is called the Analytic Server.

You are prompted by the Login dialog box to connect to an Integration Server and an OLAP Metadata Catalog. Optionally, in the Login dialog box, you can also connect to an instance of Analytic Server and create default connection settings for both server components.

For information on troubleshooting connections to server components, see the *Integration Services Administrator's Guide*

Connecting to Integration Server

The Integration Server (which is the server component of DB2 OLAP Integration Server) manages access to OLAP Metadata Catalogs and to data sources. A catalog stores information that defines the external data source and the structure of OLAP model and metaoutline elements. To create an OLAP model or a metaoutline, you must connect to Integration Server.

The Integration Server system administrator must provide a user name and password to an OLAP Metadata Catalog before you can connect to the server.

To connect to Integration Server:

1. To display the Login dialog box, from the OLAP Model main window or the Integration Services Console, select **Connections > OLAP Metadata Catalog > Connect**.

The Login dialog box is displayed automatically when you start Integration Services Console. You can redisplay the Login dialog box by using the procedures described in this topic.

2. In the **Server** drop-down list under **Essbase Integration Services**, type or select the name of an Integration Server computer; for example, cypress.

If the port number on which DB2 OLAP Integration Server communicates with the console has been set to a value other than the default, you must type the server name or IP address and type the nonstandard port number, separated by a colon; for example:

cypress:3390

3. In the **OLAP Metadata Catalog** drop-down list, select the name of an OLAP Metadata Catalog; for example, TBC_MD for the sample TBC (The Beverage Company) database.

Oracle OCI: If you are using Oracle Call Interface (OCI) direct connection, you must specify the data source name in the format, Oracle:SID; for example: Oracle:TBC.

4. In the **Code Page** drop-down list box, select the code page of the language you want to use during the current Integration Services Console session.

The code page is a portion of the locale which identifies the local language and cultural conventions such as the formatting of currency and dates and the sort order of data.

5. In the **User Name** drop-down list box under **Essbase Integration Services**, type or select your user name.
It is necessary to type a user name the first time it is used. After a user is successfully connected to Integration Services, the user name is then displayed in the **User Name** drop-down list.
6. In the **Password** text box under **Essbase Integration Services**, type your password.
Use the same user name and password that you use to connect directly to the database server and to access the database that contains the OLAP Metadata Catalog.
7. Click **OK** or enter information for connecting to an instance of Analytic Server; see “Connecting to Analytic Server.”

Connecting to Analytic Server

Analytic Server (which is the server component of DB2 OLAP Server) creates and manages OLAP databases. You do not have to connect Integration Services Console to Analytic Server to create an OLAP model or metaoutline, but you must connect to Analytic Server to view a sample OLAP outline or to load members or data into an OLAP database.

In the Login dialog box, you can specify the information for connecting to Analytic Server, but Integration Services Console does not connect to Analytic Server until you attempt to view a sample outline or to load members or data.

If you plan to use the Integration Services Console on a client computer that is outside the firewall for your network, and the console requires that Integration Server and Analytic Server be located inside the firewall for your network, you must log on to Analytic Server with a name or an IP address that connects from both sides of the firewall. The system administrator provides you with this name or IP address.

The system administrator for Analytic Server must provide you with a user name and password before you can connect.

To set connection information for an instance of Analytic Server:

1. In the OLAP Metaoutline main window, select **Connections > OLAP Metadata Catalog > Connect** to display the Login dialog box.
The Login dialog box is displayed automatically when you start Integration Services Console. You can redisplay the Login dialog box at any time while working in the OLAP Metaoutline main window by using the procedures described in this topic.
2. In the **Server** text box under **Essbase Analytic Server**, type or select the name of a computer that is running Analytic Server; for example, `sequoia`.
If you are using Integration Services Console on a client computer that is outside the firewall for your network, and you require access to instances of Integration Server and Analytic Server located inside the firewall for your network, be sure to use a name or an IP address for the instance of Analytic Server that is accessible from both sides of the firewall.
3. In the **User Name** drop-down list box under **Essbase Analytic Server**, type or select your DB2 OLAP Server user name. It is necessary to type a user name the first time it is used. After a user is successfully connected to Analytic Server, the user name is then displayed in the User Name drop-down list.

4. In the **Password** text box under **Essbase Analytic Server**, type your DB2 OLAP Server password.
5. Click **OK**.

Setting connection defaults

If you typically use a specific instance of Integration Server, OLAP Metadata Catalog, and Analytic Server, you can save the settings as the default. After you set default connection information, you can reuse these settings without reentering the information.

If you use multiple instances of Integration Server and OLAP Metadata Catalogs, you can also set multiple relationships and select the appropriate one to use so that you do not need to reenter the information.

To set connection defaults:

1. To display the Login dialog box, in the OLAP Model main window or OLAP Metaoutline main window, select **Connections > OLAP Metadata Catalog > Connect**.
2. In the Login dialog box, click **Set Login Defaults**. The Set Login Defaults dialog box is displayed.
3. In the **Server** text box, type or select the name of an Integration Server computer; for example, cypress.
4. In the **OLAP Metadata Catalog** text box, type or select the name of an OLAP Metadata Catalog; for example, TBC_MD for the sample TBC database.
5. In the **Code Page** drop-down list box, select the code page of the language you want to use for this Integration Services Console session. The code page is a portion of the locale which identifies the local language and cultural conventions such as the formatting of currency and dates and the sort order of data.
6. In the **User Name** text box, type or select a user name for the OLAP Metadata Catalog.
7. Take one of the following actions:
 - To set the connection defaults, click **Set Default**.
 - To save the connection information without setting the information as the default, click **Add to List**.
8. In the **Default Server text box**, type or select the name of an Analytic Server computer; for example, cypress.
9. In the **Default User Name** text box, type or select a user name for the instance of Analytic Server; for example, TBC.
10. Click **OK** to return to the Login dialog box.

Connecting to data sources

A data source is an external data repository, such as large database, whose data you want to analyze by using DB2 OLAP Server. You must connect DB2 OLAP Integration Server to all data sources that you plan to use in creating OLAP models and metaoutlines and in loading data into an OLAP database.

You can connect to more than one relational data source to create OLAP models.

To connect to a data source:

1. In the Integration Services Console Welcome dialog box, click the appropriate icon to create a new OLAP model or metaoutline. Alternatively, select the **Existing** or **Recent** tab and double-click an OLAP model or metaoutline to open it for editing. The Data Source dialog box is displayed.
2. In the **Data Source** drop-down list, select the data source to be used; for example, TBC in the sample application.

An ODBC data source must be created on the computer that is running Integration Server for any external data sources that you want to use. With the exception of Oracle Call Interface (OCI) direct connection, if the data source that you need is not visible in the scroll list, contact the Integration Services system administrator.

For more information about troubleshooting server and data source connections, see the *Integration Services System Administrator's Guide*.

Oracle OCI: If you are using Oracle OCI direct connection, be sure to prefix the Net Service Name (found in the `tnsnames.ora` file) or the Schema Name (if you are using `onames`) with Oracle (by typing: `Oracle:`) and then the Net Service Name or Schema Name.

The following example is a Net Service Name stanza that defines TBC in the `tnsnames.ora` file:

```
TBC =
(DESCRIPTION =
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP)(HOST = labmachine2)(PORT = 1521))
)
(CONNECT_DATA =
(SERVICE_NAME = orasid)
)
)
```

For Oracle using `onames`, in the example above, TBC is the Schema Name used to connect to an Oracle database. This is the database identifier that you use when you are using SQL *Plus to connect to a database.

DataDirect Drivers: If you are using DataDirect drivers with Oracle, pick a data source name from the **Data Source** drop-down list.

3. In the **Code Page** drop-down list box, select the code page of the language you want to use during the current Integration Services Console session, and click **OK**. The code page is a portion of the locale which identifies the local language and cultural conventions such as the formatting of currency and dates and the sort order of data.
4. In the **User Name** drop-down list box, type or select your user name.
5. In the **Password** text box, type your password.
6. Click **OK**.
7. The left frame of the OLAP Model main window initially displays information about the first data source to which you connected. Data source information is displayed hierarchically by data source name and owner name.
8. For each owner name, data is further sorted and organized by tables, views, and synonyms. You can use the listed source tables to create an OLAP model or use additional source tables by connecting to other data sources.
9. Information about subsequent data sources to which you connect is displayed in the same manner as information about the first connected data source was displayed; that is, by data source name and owner name, then tables, views, and synonyms.
10. Expand the plus sign, +, to display tables, views, and synonyms contained in the data source.

If you want to connect to additional data sources, complete the steps in the following procedure:

1. Select **Connections > Add Data Sources**. The Data Source dialog box is displayed. You can connect to any number of available data sources without closing this dialog box.
2. In the **Data Source** drop-down list, select the additional data source to which you want to connect.
Oracle OCI: If you are using OCI direct connection, be sure to prefix the Net Service Name or the Schema Name with Oracle by typing `Oracle:` and then the data source name.
If you are using DataDirect drivers with Oracle, pick a data source name from the Data Source drop-down list.
3. In the **Code Page** drop-down list box, select the code page of the language you want to use during the current Integration Services Console session, and click **OK**.
4. In the **User Name** drop-down list box, type or select your user name. It is necessary to type a user name the first time it is used. After a user is successfully connected to a data source, the user name is then displayed in the **User Name** drop-down list.
5. In the **Password** text box, type your password and click **Connect**. The left frame of the OLAP Model main window displays information about the data source to which you just connected along with information about the first data source to which you connected.
6. Repeat step 2 through step 3 for each data source to which you want to connect. As you connect to additional data sources, the left frame of the OLAP Model main window displays information about all data sources to which you are connected.
7. When you finish connecting to all appropriate data sources, click **Close**.

Reconnecting when connections are lost

If a connection is lost to Integration Server and OLAP Metadata Catalog, you can reconnect without losing any unsaved work on the OLAP models or metaoutlines.

To reconnect to Integration Server and OLAP Metadata Catalog:

1. Start Integration Server using one of the methods described in “Starting the Integration Server on Windows” on page 26 or “Starting Integration Server on UNIX” on page 41.
2. In the Integration Services Console, select **Connections > OLAP Metadata Catalog > Reconnect**.

The connection is made automatically. No messages are displayed, and you do not need to reenter information in the Login dialog box.

Chapter 8. Configuring relational data sources for DB2 OLAP Integration Server

To create OLAP models or build metaoutlines, you must connect DB2 OLAP Integration Server to a database that contains metadata (the OLAP Metadata Catalog database) and to a database in which user data is stored (the Data Source database). To make these connections, you must configure a data source name for an OLAP Metadata Catalog database and a Data Source database. You do this by mapping a supported Open Database Connectivity (ODBC) driver to each database.

Note: You are not required to use two different databases for your Data Source database and OLAP Metadata Catalog database. Additionally, you are not required to use two different data source names to connect to the Data Source database and OLAP Metadata Catalog database if the two databases reside in the same database.

On Windows, use the ODBC Administrator to configure ODBC data sources. On UNIX, edit the `odbc.ini` file using a text editor such as `vi`. You cannot use the `odbcconfig` utility because it is no longer included in DB2 OLAP Integration Server. All ODBC data source names are configured only on the computer that runs DB2 OLAP Integration Server.

Note: If a supported ODBC driver is already mapped to a Data Source database, you do not need to map it again. In this case, you need to map a supported ODBC driver only to the OLAP Metadata Catalog database.

If you are using Oracle OCI direct connection, you connect Integration Services to an OLAP Metadata Catalog database and to a Data Source database when you access the databases in Integration Services Console. This process is described in “Connecting to Integration Server” on page 58 and “Connecting to data sources” on page 60.

This chapter contains the following topics that describe how to configure data sources:

- “About data source configuration” on page 63
- “Configuring data source names on Windows” on page 64
- “Using Oracle OCI Direct Connection for Oracle” on page 71
- “Configuring data source names on UNIX systems” on page 72
- “After configuration of data source names” on page 78

About data source configuration

In DB2 OLAP Integration Server, a data source must be configured for the relational database that contains user data (the Data Source database). A data source must also be configured for the relational database that contains the OLAP Metadata Catalog. Databases, including an OLAP Metadata Catalog, can run on any supported platform, provided that you have the ODBC driver and, in some cases, the database client software needed to access them.

The installation program installs DataDirect ODBC driver files that you use to configure data source names for both the Data Source database and OLAP Metadata Catalog database.

If database client software is required, the relational database management system (RDBMS) should include one of the database clients listed in “Supported ODBC drivers and relational data sources” on page 14. If your configuration requires database client software, ensure that the versions of the database client and the RDBMS are compatible.

This chapter provides information on how to configure ODBC drivers. For information on how to install and configure database client software, see the RDBMS documentation.

Server and client software and ODBC connections

Integration Server runs on Windows 2000, Windows XP, Windows 2003, AIX, HP-UX, Solaris, or Linux. You must install Integration Server software, including ODBC drivers, and if required, database client software on the server computer.

The DB2 OLAP Integration Server client software runs on Windows 2000, Windows XP, or Windows 2003. You must install the Integration Services Console software on Integration Services client computers.

Configure the ODBC connections to the Data Source database and the OLAP Metadata Catalog database only on the computer running the server. You do not need to configure ODBC connections on client computers.

If you are using Oracle OCI direct connection, you connect DB2 OLAP Integration to both an OLAP Metadata Catalog database and a Data Source database when you access the databases in Integration Services Console. This process is described in “Connecting to Integration Server” on page 58 and “Connecting to data sources” on page 60.

Supported ODBC drivers

For a complete list of supported ODBC drivers by platform, see “Supported ODBC drivers and relational data sources” on page 14.

For detailed information on the DataDirect Wire Protocol drivers provided with Integration Services, see the *DataDirect Connect for ODBC Reference*.

Configuring data source names on Windows

To configure data source names on Windows systems, you must start ODBC Administrator and then map an ODBC driver to one or more relational databases that you will use for creating and storing OLAP models and metaoutlines. Run the ODBC Administrator utility from the Windows Control Panel.

The procedures in the following sections show you how to configure data source names to create connections to databases. The examples in the procedures use a database named TBC as the sample Data Source database and a database named TBC_MD as the sample OLAP Metadata Catalog database.

For each site-specific data source name that you configure, obtain the following information from your database administrator:

- The name of the relational database for which you want to configure a data source name

For example, in the sample application, TBC is the name of the Data Source database; TBC_MD is the name of the OLAP Metadata Catalog database.

- IP address or name of the computer on which the database tables are stored
You can enter either an alphabetic computer name (for example, sequoia), or an IP address (for example, 172.0.0.125).

In the sample application, if you are configuring a data source name for the TBC database, use the IP address of the computer where the TBC database tables are stored. If you are configuring a data source name for the TBC_MD database (the sample OLAP Metadata Catalog database), use the IP address of the computer where the TBC_MD tables are stored.

- Port number on which your database listens

For clarity in presenting the sample application in Integration Services Console, the sample Data Source database and sample OLAP Metadata Catalog database are two separate databases that use two different data source names. You are not required to use two different databases for your Data Source database and OLAP Metadata Catalog database. Additionally, you are not required to use two different data source names to connect to your Data Source database and OLAP Metadata Catalog database if all the tables reside in the same database.

Note: If you are using Oracle OCI direct connection, you do not need to configure data sources. You connect DB2 OLAP Integration Server to relational data sources when you access them in Integration Services Console. This process is described in “Connecting to Integration Server” on page 58 and “Connecting to data sources” on page 60.

Accessing the ODBC Data Source Administrator

For any RDBMS, you must access the Windows ODBC Data Source Administrator to begin the data source name configuration process.

To access the Administrator:

1. On the Windows desktop, select **Start** —> **Settings** —> **Control Panel** to open the **Control Panel** window.
2. In the **Control Panel** window, double-click the **Administrative Tools** icon, and then double-click the **Data Sources (ODBC)** icon to open the **ODBC Data Source Administrator** dialog box.
3. In the **ODBC Data Source Administrator** dialog box, click the **System DSN** tab.

Configuring DataDirect Wire Protocol drivers

Use the following procedures to configure a data source name for your Data Source database using a DataDirect Wire Protocol driver for DB2 Universal Database™, Informix, Oracle, or Sybase. Then repeat the procedure to configure a data source name for the relational database that you want to use as your OLAP Metadata Catalog database.

Before beginning the configuration procedure, perform the procedure in “Accessing the ODBC Data Source Administrator” on page 65.

Note: You may repeat the configuration procedure to create any number of data source names to relational Data Source databases or OLAP Metadata Catalog databases.

To configure a data source name:

1. Click **Add** to open the **Create New Data Source** dialog box.
2. In the driver list box of the **Create New Data Source** dialog box, select the appropriate Wire Protocol driver for your RDBMS, and click **Finish**.
The **Wire Protocol Driver Setup** dialog box for your RDBMS is displayed.
3. In the **Data Source Name** text box, type the name that you want to use for this data source connection.

In the sample application, TBC is the data source name used to connect to the Data Source database called TBC; TBC_MD is the data source name used to connect to the sample OLAP Metadata Catalog database called TBC_MD.

4. **Optional:** If you want to record a description, in the **Description** text box, type a description that indicates how you use this data source name.

For example, you might type the following to describe the My Business database:

Customers, products, markets

You might type the following to describe the sample application database:

Sample relational data source

You might type the following to describe the sample OLAP Metadata Catalog database:

Sample OLAP Metadata Catalog

5. Make the necessary entries to configure data source names by completing the set of steps appropriate to your RDBMS:

DB2 Universal Database or DB2 UDB for z/OS and OS/390:

- a. In the **IP Address** text box, type the address (either the computer name or the numeric IP address) of the computer where the database tables are stored.
- b. In the **TCP Port** text box, type the port number on which the DB2 database server listens.
 - If you are using DB2 UDB, the default port number is 50000.
 - If you are using DB2 UDB for z/OS and OS/390, the default port number is 446.
- c. If you are using DB2 UDB, in the **Database Name** text box, type the name of the database for which you want to create this data source name.
- d. If you are using DB2 UDB for z/OS and OS/390, perform the following actions:
 - 1) In the **Location** text box, type the DB2 location name as defined during the DB2 UDB for z/OS and OS/390 installation.
 - 2) In the **Collection** text box, type the name that identifies a logical group of database objects (the default is DATADIRECT00).

Note: For DB2 UDB for z/OS and OS/390, you configure a data source name for the Data Source database only. You cannot configure a data source name for the OLAP Metadata Catalog database. OLAP Metadata Catalog functionality is not supported in DB2 UDB for z/OS and OS/390.

To use DB2 OLAP Integration Server with DB2 UDB for z/OS and OS/390, you must add an entry to the `eis.cfg` file after you configure the data source name for the source database. See “Adding a configuration file entry for DB2 UDB for z/OS and OS/390” on page 68 for more information.

- e. Click the **Bind** tab.
- f. Accept the default in the **Bind** tab and click the **Create Package** button.
For more information about the bind process, see *DataDirect Connect for ODBC Reference*.
- g. In the **Login** dialog box, type the user name and password associated with the database for which this ODBC data source name is being created, and click **OK**.

A message is displayed informing you that the package was created successfully.

Informix:

- a. In the **Host Name** text box, type the address (either the computer name or the numeric IP address) of the computer where the database tables are stored.
- b. In the **Port Number** text box, type the port number on which the Informix database server listens.
The default port number for Informix is 1526.
- c. In the **Server Name** text box, type the address, either numerically or by name, of the Informix server as it appears in the `sqlhosts` file.
If you do not know this information, contact your Informix database administrator.
- d. In the **Database Name** text box, type the name of the database for which you want to create this data source name.

Oracle:

- a. In the **Host** text box, type the address (either the computer name or the numeric IP address) of the computer where the database tables are stored.
- b. In the **Port Number** text box, type the port number on which the Oracle database server listens.
The default port number for Oracle is 1521.
- c. In the **SID** text box, type the name of the database for which you want to create this data source name.

Sybase:

- a. In the **Network Address** text box, type the address (either the computer name or the numeric IP address) of the computer where the database tables are stored and the port number on which the Sybase database server listens. Separate the computer name or IP address and the port number with a comma as shown in the following examples:
By name: `sequoia,4100`
Numerically: `172.0.0.125,4100`
 - b. In the **Database Name** text box, type the name of the database for which you want to create this data source name.
6. Test the connection by following the procedures in “Testing DataDirect Wire Protocol drivers” on page 68.

When you finish testing, repeat 2 through 5 to configure a data source name for an OLAP Metadata Catalog database.

Testing DataDirect Wire Protocol drivers

To test a data source name connection:

1. Click **Test Connect**.

The **Login to Wire Protocol** dialog box for your RDBMS is displayed.

2. In the **User Name** and **Password** text boxes, enter a valid user name and password for this data source connection, and click **OK**.

If an error message is displayed, make the necessary corrections, and click **Test Connect** again.

When no errors occur, a dialog box displays the message, **Connection established!**

3. In the message dialog box, click **OK**.

You are returned to the **ODBC Wire Protocol Driver Setup** dialog box.

4. In the **ODBC Wire Protocol Driver Setup** dialog box, click **OK**.

You are returned to the **System DSN** tab of the **ODBC Data Source Administrator** dialog box. The data source name that you entered and the driver that you mapped to the data source name are displayed in the **System Data Sources** list box.

5. Repeat steps 1 through 5 in “Configuring DataDirect Wire Protocol drivers” on page 65 to configure a data source name for an OLAP Metadata Catalog database.
6. When you have completed configuration of all needed data source names, click **OK** to close the **ODBC Data Source Administrator** dialog box.

Adding a configuration file entry for DB2 UDB for z/OS and OS/390

If you are using IBM DB2 UDB for z/OS and OS/390, you must add an entry to the `eis.cfg` file to support DB2 OLAP Integration Server functionality. You must add entries for each data source name for DB2 UDB for z/OS and OS/390 that you configured in “Configuring DataDirect Wire Protocol drivers” on page 65.

Note: For DB2 UDB for z/OS and OS/390, you configure a data source name for the Data Source database only. You cannot configure a data source name for the OLAP Metadata Catalog database. OLAP Metadata Catalog functionality is not supported in DB2 UDB for z/OS and OS/390.

To add an entry for DB2 UDB for z/OS and OS/390 to the `eis.cfg` file:

1. On the computer running Integration Server, use a text editor to open `eis.cfg`.
2. Add a new line for DB2 UDB for z/OS and OS/390 using the data source name or names you configured in “Configuring DataDirect Wire Protocol drivers” on page 65.

Add separate lines for each data source name configured for DB2 UDB for z/OS and OS/390. The syntax is:

```
[DS:dsn:390]
```

For example, using the sample application database, TBC, the following is the entry to the `eis.cfg` file:

```
[DS:TBC:390]
```

3. Save and close `eis.cfg`.

Configuring the SQL Server ODBC Driver for Microsoft SQL Server

Use the following procedure to configure a data source name for your Data Source database using the SQL Server driver. Then repeat the procedure to configure a data source name for your OLAP Metadata Catalog database.

Note: You may repeat the configuration procedure to create any number of data source names to relational Data Source databases or OLAP Metadata Catalog databases.

The procedures that follow provide information on basic configuration. If you are using more advanced options, refer to the online help for each wizard screen for assistance in completing the wizards.

To use the SQL Server driver to configure a data source name:

1. Access **ODBC Data Source Administrator** by following the procedure in “Accessing the ODBC Data Source Administrator” on page 65.
2. Click **Add** to open the **Create New Data Source** dialog box.
3. In the driver list box of the **Create New Data Source** dialog box, select the SQL Server driver, and click **Finish**.

The **Create a New Data Source to SQL Server** dialog box is displayed.

4. In the **Name** text box, type the data source name that you want to use for this data source connection.

In the sample application, TBC is the data source name used to connect to the Data Source database called TBC; TBC_MD is the data source name used to connect to the sample OLAP Metadata Catalog database called TBC_MD.

5. **Optional:** In the **Description** text box, type a description that indicates how you use this data source name.

For example, you might type the following names to describe the My Business database:

Customers, products, markets

You might type the following statement to describe the sample application database:

Sample relational data source

You might type the following statement to describe the sample OLAP Metadata Catalog database:

Sample OLAP Metadata Catalog

6. In the **Server** text box, type the address (either the computer name or the numeric IP address) of the computer where the database tables are stored.
7. Click **Next**.
The second wizard screen is displayed.
8. Select the appropriate option button for the method to use for verifying login IDs:
 - **With Window NT authentication using the network login ID.**
 - **With SQL Server authentication using a login ID and password entered by the user.**
9. **Optional:** If you are using a network library other than TCP/IP (the default) to connect to the database server, perform the following steps:
 - a. Click the **Client Configuration** button.

- b. In the **Network libraries** list, select the appropriate option button for the library that you are using to connect to the database server computer, and click **OK**.

You are returned to the second wizard screen.

10. **Optional:** Select the **Connect to SQL Server to obtain default settings for the additional configuration** check box.
When this check box is selected, the driver obtains default settings from SQL Server that it uses to complete additional setup screens in the wizard.
When this check box is clear, the driver uses standard defaults to complete the additional setup screens in the wizard.
11. Click **Next**.
The third wizard screen is displayed.
12. Select the **Change the default database to** check box and then type or select the name of the database for which you want to create this data source name.
For example, in the sample application, TBC is the name of the Data Source database; TBC_MD is the name of the OLAP Metadata Catalog database.
13. Click **Next**.
The fourth wizard screen is displayed.
14. Make any entries appropriate to your system configuration and click **Finish**.
15. Repeat steps 2 through 14 to configure a data source name for an OLAP Metadata Catalog.
16. When you have completed configuring all needed data sources, click **OK** to close the **ODBC Data Source Administrator** dialog box.

Configuring the Teradata ODBC driver for Teradata

Use the following procedure to configure a data source name for your Data Source database using the Teradata driver. Then repeat the procedure to configure a data source name for the relational database that you want to use as your OLAP Metadata Catalog database.

Note: You may repeat the configuration procedure to create any number of data source names to Data Source databases or OLAP Metadata Catalog databases.

To configure a data source name:

1. Access the **ODBC Data Source Administrator** by following the procedure in “Accessing the ODBC Data Source Administrator” on page 65.
2. Click **Add** to open the **Create New Data Source** dialog box.
3. In the driver list box of the **Create New Data Source** dialog box, select Teradata, and click **Finish**.
The **ODBC Driver Setup for Teradata RDBMS** dialog box is displayed.
4. In the **Name** text box, type the data source name that you want to use for this data source connection.
In the sample application, TBC is the data source name used to connect to the Data Source database called TBC; TBC_MD is the data source name used to connect to the sample OLAP Metadata Catalog database called TBC_MD.
5. **Optional:** In the **Description** text box, type an description that indicates how you use this data source name.
For example, you might type the following to describe the My Business database:

Customers, products, markets

You might type the following to describe the sample application database:

Sample relational data source

You might type the following to describe the sample OLAP Metadata Catalog database:

Sample OLAP Metadata Catalog

6. In the **Teradata Server Info** text box, type the address (either the computer name or the numeric IP address) of the computer where the database tables are stored.
7. In the **Default Database** text box, type the name of the database for which you want to create this data source name.
For example, in the sample application, TBC is the name of the Data Source database; TBC_MD is the name of the OLAP Metadata Catalog database.
8. **Optional:** Enable the X Views option for Teradata.
You can limit the number of data sources displayed in the left frame of an OLAP Model main window based on the security of the Teradata user ID that was used to connect. Follow steps a through c to enable the X Views option:
 - a. Click the **Options** button.
 - b. Check the **Use X Views** check box.
 - c. Click **OK** to return to the **ODBC Driver Setup for Teradata RDBMS** dialog box.
9. Click **OK** to return to the **System DSN** tab of the **ODBC Data Source Administrator** dialog box.
The data source name that you entered and the driver that you mapped to it are displayed in the **System Data Sources** list box.
10. Repeat steps 2 through 9 above to configure a data source name for an OLAP Metadata Catalog.

Note: You may repeat the above procedure to create any number of data source connections to relational data sources or OLAP Metadata Catalogs.
11. When you have completed configuring all needed data sources, click **OK** to close the **ODBC Data Source Administrator** dialog box.

Editing a Data Source Name

To edit configuration information for a data source name:

1. Access the **ODBC Data Source Administrator** by following the procedures in “Accessing the ODBC Data Source Administrator” on page 65.
2. Select the data source name and click **Configure** to open the driver setup dialog box or wizard specific to your RDBMS.
3. Correct any information that you want to change.

Using Oracle OCI Direct Connection for Oracle

If you are using Oracle OCI, DB2 OLAP Integration Server requires that the Oracle client software must be 32-bit. Oracle 32-bit client software and the server component of DB2 OLAP Integration Server must be installed on the same computer. When using OCI direct connection, the database client software is required to make connections to RDBMS databases. For information about installing and configuring database client software, refer to the Oracle RDBMS installation documentation.

On Windows systems, environment changes needed for running the database client software should be completed automatically by the Oracle 32-bit client installation program. For information about additional configuration of the database client software, see the Oracle documentation.

On UNIX systems, you must manually configure environment variables to access database client software and libraries. The variable settings must be defined for the environment of the user account (for example, `hyperion`) that you use to run the server. For information on setting the UNIX environment to access a database client, see “Updating the environment for the database client” on page 38.

To test OCI connections, make sure that you can connect to RDBMS databases and access tables while logged on with the user account that you use to run Integration Server. Use the SQL command line utility provided with the Oracle 32-bit database client software that you are using.

With Oracle OCI, you connect to the OLAP Metadata Catalog database and Data Source database at login time in Integration Services Console. You make the connection by typing `Oracle:` before the data source name for the OLAP Metadata Catalog database and the data source name for the relational database. This process is described in “Connecting to Integration Server” on page 58 and “Connecting to data sources” on page 60.

Configuring data source names on UNIX systems

Neither the ODBC Administrator or the `odbcconfig` utility is available on UNIX systems. On UNIX, you must edit the `odbc.ini` file to set environment variables, to configure, add, or change data source names, and to add or change drivers.

If you are using Oracle OCI direct connection, you do not need to configure data source names on UNIX. You connect Integration Services to both an OLAP Metadata Catalog database and a Data Source database when you access them in Integration Services Console. This process is described in “Connecting to Integration Server” on page 58 and “Connecting to data sources” on page 60.

Configuring environment variables

On UNIX systems, you must set environment variables to enable access to ODBC core components. See “Configuring the environment on UNIX” on page 36 for information about using shell scripts created by the installation program to set the required variables.

Oracle OCI: If you are using Oracle OCI drivers, you must set environment variables for the database client software as described in “Updating the environment for the database client” on page 38.

Teradata: If you are using Teradata drivers, you must set environment variables to point to the location of the installed drivers as specified in “Updating the environment for the database client” on page 38.

Note: Use the `env` command to verify environment settings.

Editing the `odbc.ini` file

You configure data source names for a Data Source database or OLAP Metadata Catalog database in an `odbc.ini` file. In one section of the file, add a name and

description for the ODBC data source. In a separate, newly created section of the file, provide the ODBC driver path, file name, and all other required driver settings.

The installation program installs a sample `odbc.ini` file in the `HYPERION_HOME/common/ODBC/Merant/4.2` directory. The file contains generic ODBC connection and configuration information for supported ODBC drivers. Use the file as a starting point to map the ODBC drivers that you use to the Data Source database and OLAP Metadata Catalog database.

DB2 Universal Database: For each server computer to which you are connecting using the DataDirect Wire Protocol driver, you must create a bind package. DataDirect supplies a program that creates the bind package for you. See “Creating DB2 bind packages for DataDirect Wire Protocol driver connections” on page 77 for instructions on using the program supplied by DataDirect.

Teradata: When configuring a data source name, the data source description must be set to `tdata.sl` (HP-UX) or `tdata.so` (AIX and Solaris). Specific examples for AIX, HP-UX, and Solaris operating systems are shown in “Examples of ODBC settings for Teradata” on page 76

Note: If you use a file other than the `odbc.ini` file (located by default in `$HYPERION_HOME/common/ODBC/Merant/4.2`), be sure to set the `ODBCINI` environment variable to the name of the file that you use.

To add an ODBC data source to an `odbc.ini` file:

1. On the computer running Integration Server, open the `odbc.ini` file by using a text editor such as `vi`.
2. Locate the section starting with `[ODBC Data Sources]` and add a new line with the data source name and description; for example:

```
mydata=data source for analysis
```

To minimize confusion, the data source name can match the name of the database in the RDBMS.

3. Add a new section to the file by creating a new line with the new data source name enclosed in brackets; for example:

```
[mydata]
```

4. On the lines following the data source name, add the full path and file name for the ODBC driver required for this data source and any other required ODBC driver information.

Use the examples shown in the following sections as guidelines for specific RDBMSs.

Note: Make sure that the ODBC driver file actually exists in the location that you specify for the `Driver=` setting.

5. When you finish editing `odbc.ini`, save the file and exit the text editor.

For information about the `odbc.ini` file and the ODBC driver settings for each RDBMS, see the *DataDirect Connect for ODBC Reference* in PDF format and the `Readme` files in the `$HYPERION_HOME/common/ODBC/Merant/4.2/docs` directory. This manual is also installed with the DB2 OLAP Server documentation. For information about vendor-supplied ODBC driver settings, refer to the installation documentation for the vendor-supplied ODBC drivers.

Examples of ODBC settings for DB2 Universal Database

The following example illustrates how you might edit `odbc.ini` to connect to a Data Source database, such as `db2data` on DB2 UDB Version 7.1 (on AIX), using a DataDirect Wire Protocol driver. You can make a separate entry to connect to the OLAP Metadata Catalog database.

Use the `vi $ODBCINI` command to edit the `odbc.ini` file and insert the following statements:

```
[ODBC Data Sources]
db2data=DB2 Source Data on AIX
...
[db2data]
Driver=
%HYPERION_HOME%/common/ODBC/Merant/4.2/lib/ARdb218.so
IpAddress=isaix7
Database=tbd7a7n
TcpPort=50000
Package=PACK4A
Action=REPLACE
QueryBlockSize=8
CharSubTypeType=SYSTEM_DEFAULT
ConversationType=SINGLE_BYTE
CloseConversation=DEALLOC
UserBufferSize=32
MaximumClients=35
GrantExecute=1
GrantAuthid=PUBLIC
OEMANSI=1
DecimalDelimiter=PERIOD
DecimalPrecision=15
StringDelimiter=SINGLE_QUOTE
IsolationLevel=CURSOR_STABILITY
ResourceRelease=DEALLOCATION
DynamicSections=32
Trace=0
WithHold=0
```

Example of ODBC settings for DB2 UDB for z/OS and OS/390

The following example illustrates how you might edit `odbc.ini` to connect to a Data Source database, such as `os390` on DB2 UDB for z/OS and OS/390 Version 7, using a DataDirect Wire Protocol driver. You can make a separate entry to connect to the OLAP Metadata Catalog database.

Use the `vi $ODBCINI` command to edit the `odbc.ini` file and insert the following statements:

```
[ODBC Data Sources]
os390= Source data on DB2 UDB for z/OS and OS/390
...
[os390]
Driver=
%HYPERION_HOME%/common/ODBC/Merant/4.2/lib/ARdb219.so
Collection=DATADIRECT00
Database=
DynamicSections=100
GrantAuthid=PUBLIC
GrantExecute=1
IpAddress=isaix21
IsolationLevel=CURSOR_STABILITY
Location=DALLAS
Package=PACK42
PackageOwner=
TcpPort=446
WithHold=1
```

Note: To use DB2 OLAP Integration Server with DB2 UDB for z/OS and OS/390, you must add an entry to the `eis.cfg` file after you configure the data source name for the source database. See “Adding a configuration file entry for DB2 UDB for z/OS and OS/390” on page 77 for more information.

Example of ODBC settings for Informix

The following example illustrates how you might edit `odbc.ini` to connect to a Data Source database, such as `ifxdata` on Informix Version 9.3 (on HP-UX), using a DataDirect Version 4.2 ODBC driver. You can make a separate entry to connect to the OLAP Metadata Catalog database.

```
[ODBC Data Sources]
ifxdata=Informix 9.3 Source Data on HP-UX
...
[ifxdata]
Driver=
%HYPERION_HOME%/common/ODBC/Merant/4.2/lib/ARifc118.so
Description=Informix9.3 ODBC Driver
Database=ifxdata
HostName=ishp4
PortNumber=1526
ServerName=informixhost
```

Example of ODBC settings for Oracle

If an installation uses Oracle `onames` instead of `tnsnames`, use the Oracle OCI interface instead of ODBC and connect directly to relational data sources when they are accessed in Integration Services Console.

The following example illustrates how you might edit `odbc.ini` to connect to a Data Source database, `oradata`, on Oracle 9i (on Solaris), using a DataDirect Version 4.2 ODBC driver. You can make a separate entry to connect to the OLAP Metadata Catalog database.

```
[ODBC Data Sources]
oradata=Oracle9i Source Data on Solaris
...
[oradata]
Driver=
%HYPERION_HOME%/common/ODBC/Merant/4.2/lib/ARora18.so
Description=my oracle source
HostName=oraclehost
SID=tbc1
PortNumber=1521
```

Note: Run `ivtestlib` to verify that the environment is set to run the correct ODBC driver file. For example, run `ivtestlib` and paste the path and file name that follow `Driver=` in the `odbc.ini` file that you edited.

Example of ODBC settings for SQL Server

The following example illustrates how you might edit `odbc.ini` to connect to a Data Source database, such as `sqldata`, on SQL Server 2000 (on HP-UX), using a vendor-supplied ODBC driver. You can make a separate entry to connect to the OLAP Metadata Catalog database.

```
[ODBC Data Sources]
sqldata=SQL Server 2000 Source Data on HP-UX
...
[sqldata]
Driver=
%HYPERION_HOME%/common/ODBC/Merant/4.2/lib/ARmsss18.so
Description=SQL Server 2000 Driver
```

```
Database=sqldata
Address=isqa17,1433
QuotedId=No
AnsiNPW=No
```

Example of ODBC settings for Sybase

The following example illustrates how you might edit `odbc.ini` to connect to a Data Source database, such as `sybdata`, on Sybase Version 12 (on HP-UX), using a DataDirect Version 4.2 ODBC driver. You can make a separate entry to connect to the OLAP Metadata Catalog database.

```
sybdata=Sybase 12.0 Source Data on HP-UX
...
[sybdata]
Driver=
%HYPERION_HOME%/common/ODBC/Merant/4.2/lib/ARase18.so
Description=Sybase 12.0 ODBC Driver
Database=sybdata
NetworkAddress=isso15,4200
```

Examples of ODBC settings for Teradata

AIX: The following example illustrates how you might edit `odbc.ini` to connect to a Data Source database, `tbc`, on Teradata (on AIX), using the Teradata ODBC driver. You can make a separate entry to connect to the OLAP Metadata Catalog database. The data source description in [ODBC Data Sources] must be set to `tdata.so`.

```
[ODBC Data Sources]
tbc=tdata.so
...
[tbc]
Driver=/usr/odbc/drivers/tdata.so
Description=NCR 3600 running Teradata V2R4
DBCName=139.64.140.45
DefaultDatabase=tbc
UseXViews=Yes
```

HP-UX: The following example illustrates how you might edit `odbc.ini` to connect to the same Data Source database, `tbc`, on Teradata (on HP-UX), using the Teradata ODBC driver. You can make a separate entry to connect to the OLAP Metadata Catalog database. The data source description in [ODBC Data Sources] must be set to `tdata.sl`.

```
[ODBC Data Sources]
tbc=tdata.sl
...
[tbc]
Driver=/usr/odbc/drivers/tdata.sl
Description=NCR 3600 running Teradata V2R4
DBCName=139.64.140.45
DefaultDatabase=tbc
UseXViews=Yes
```

Solaris: The following example illustrates how you might edit `odbc.ini` to connect to the same Data Source database, `tbc`, on Teradata (on Solaris), using the Teradata ODBC driver. You can make a separate entry to connect to the OLAP Metadata Catalog database. The data source description in [ODBC Data Sources] must be set to `tdata.so`.

```
[ODBC Data Sources]
tbc=tdata.so
...
[tbc]
Driver=/usr/odbc/drivers/tdata.so
```

Description=NCR 3600 running Teradata V2R4
DBCName=139.64.140.45
DefaultDatabase=tbc
UseXViews=Yes

Creating DB2 bind packages for DataDirect Wire Protocol driver connections

In the UNIX environment, you must create a bind package for each connection between DB2 OLAP Integration Server and a DB2 UDB database that you make using the MERANT Wire Protocol driver. Without a bind package for each connection, the Wire Protocol driver will not work properly. MERANT supplies a command line program to create the bind packages.

To create a bind package for a connection using the MERANT Wire Protocol Driver for DB2 UDB:

1. On the computer running Integration Server, open a command shell window.
2. In the command shell window type:

```
bind19 dsn
```

Replace *dsn* with the data source name for which you are creating this bind package. For example, to create a bind package for the data source name TBC, type:

```
bind19 TBC
```

Note: You are prompted for a user name and password if this information is not stored in the system information file.

If successful, a message is displayed stating that the package was created and bound.

For troubleshooting information, access the DataDirect Web site at <http://www.datadirect-technologies.com> and select the Product Documentation link.

3. Repeat step 2 for all bind packages you want to create.
4. When all bind packages are created, close the command shell window.

Adding a configuration file entry for DB2 UDB for z/OS and OS/390

If you are using IBM DB2 UDB for z/OS and OS/390, you must add an entry to the `eis.cfg` file to support DB2 OLAP Integration Server functionality. You must add entries for each data source name for DB2 UDB for z/OS and OS/390 that you configured in “Configuring DataDirect Wire Protocol drivers” on page 65.

Note: For DB2 UDB for z/OS and OS/390, you configure data source name for the Data Source database only. You cannot configure a data source name for the OLAP Metadata Catalog database. OLAP Metadata Catalog functionality is not supported in DB2 UDB for z/OS and OS/390.

To add an entry for DB2 UDB for z/OS and OS/390 to the `eis.cfg` file:

1. On the computer running the server component of DB2 OLAP Integration Server, use a text editor to open `eis.cfg`.
2. Add a new line for DB2 UDB for z/OS and OS/390 using the data source name or names you configured in “Configuring DataDirect Wire Protocol drivers” on page 65.

Add separate lines for each data source name configured for DB2 UDB for z/OS and OS/390. The syntax is:

```
[DS:dsn:390]
```

For example, using the sample application database, TBC, the following is the entry to the `eis.cfg` file:

```
[DS:TBC:390]
```

3. Save and close `eis.cfg`.

After configuration of data source names

After you configure data source names for the Data Source database and OLAP Metadata Catalog database, you can connect to the databases from DB2 OLAP Integration Server. You can then create, modify, and save OLAP models and metaoutlines. For information on viewing TBC tables, columns, OLAP models, and metaoutlines in Integration Services Console, see “Viewing TBC tables and columns” on page 106.

Note: The SQL Server ODBC driver may time out during a call to an SQL Server database, particularly during a data load. If a timeout occurs, try again when the database is not busy. Increasing the driver time-out period may avoid this problem. For more information, see the ODBC documentation for the driver that you are using.

For more information on ODBC connection problems and solutions, see the *Integration Services Administrator's Guide*.

Chapter 9. Creating, upgrading, and deleting OLAP Metadata Catalogs

Before you run DB2 OLAP Integration Server, you must create an OLAP Metadata Catalog to store the OLAP models that you create and the metaoutlines that you build from the OLAP models.

Note: If you intend to use the Sample application to acquaint yourself with Integration Services, skip to Chapter 10, “Setting up the sample application for DB2 OLAP Integration Server,” on page 87. The Sample application has its own relational data source, OLAP Metadata Catalog, OLAP model, and metaoutline. Return to this chapter when you are ready to use a relational data source to build OLAP models and metaoutlines.

It is recommended that you use the automatic installation process to install the Sample applications. See “Setting up the standard sample application automatically” on page 89.

This chapter tells you what to consider before you create an OLAP Metadata Catalog in a relational database. You can create the catalog either automatically, through the use of Integration Services Console, or manually. You can also create an Unicode-enabled OLAP Metadata Catalog. For manual creation of an OLAP Metadata Catalog, this chapter lists the SQL scripts that you must run to create tables for the catalog. The procedures in this chapter assume that you know how to create tables by running SQL scripts.

This chapter contains the following topics:

- “About OLAP Metadata Catalogs”
- “Creating or upgrading OLAP Metadata Catalogs” on page 80
- “Mapping OLAP Metadata Catalogs” on page 84
- “Deleting OLAP Metadata Catalogs” on page 84

About OLAP Metadata Catalogs

An OLAP Metadata Catalog is a database that consists of 33 tables in which DB2 OLAP Integration Server stores OLAP models and metaoutlines.

You can run the RDBMS for an OLAP Metadata Catalog on any platform as long as you have the ODBC driver and database client software required to connect to the catalog from the computer that runs DB2 OLAP Integration Server. You can also run the RDBMS for the relational data source that you use to create OLAP models and build metaoutlines on any platform, if you have the ODBC driver and database client software required to access the data source.

The RDBMS for the OLAP Metadata Catalog can be different from the RDBMS for the relational data source, and the platforms for the two RDBMSs do not need to be the same.

You can have more than one OLAP Metadata Catalog and, by using the XML Import/Export utility, you can move OLAP models and metaoutlines from one OLAP Metadata Catalog to another OLAP Metadata Catalog.

Creating or upgrading OLAP Metadata Catalogs

To create an OLAP Metadata Catalog, you must have permission to create tables in the database or have whatever similar access privileges are required by the RDBMS that you are using.

Note: On MS SQL Server, the tables in a catalog are not accessible if they are created by a user that does not have db_owner privileges.

To create an OLAP Metadata Catalog:

1. Create a database for OLAP Metadata Catalog tables using the applicable RDBMS utility.
See “Creating databases for OLAP Metadata Catalog tables.”
2. Take one of the following actions:
 - Create an OLAP Metadata Catalog automatically by using Integration Services Console (see “Creating or upgrading the OLAP Metadata Catalog automatically”).
 - Create tables for the OLAP Metadata Catalog database by running SQL scripts (see “Creating the OLAP Metadata Catalog manually” on page 81).

Creating databases for OLAP Metadata Catalog tables

Create a database for the OLAP Metadata Catalog tables by using the applicable RDBMS utility, in the same way that you create any database.

- Create a database device or tablespace.
- Allocate 30 MB for storage.
- Create user names and passwords, if needed.
- Grant user privileges or permissions.

One OLAP Metadata Catalog can store all OLAP models and metaoutlines from numerous relational data sources. You can also create more than one OLAP Metadata Catalog to store additional OLAP models and metaoutlines for different projects.

Creating or upgrading the OLAP Metadata Catalog automatically

When you complete installation of Integration Services and start Integration Services Console, the program automatically displays the OLAP Metadata Catalog Setup dialog box. This dialog box enables you to create a standard OLAP Metadata Catalog automatically, using Integration Services Console. If you have an older version of the OLAP Metadata Catalog, you also use this dialog box to upgrade the older version to the current version.

If you prefer to create the OLAP Metadata Catalog manually, see “Creating the OLAP Metadata Catalog manually” on page 81.

To create an OLAP Metadata Catalog automatically:

1. From the **Server Name** drop-down list in the **OLAP Metadata Catalog Setup** dialog box, type or select the server computer on which you have installed DB2 OLAP Integration Server.

It is necessary to type a server name the first time it is used. After the OLAP Metadata Catalog is successfully created, the server name is then displayed in the **Server Name** drop-down list box.

2. From the **Catalog ODBC DSN** drop-down list, select the Data Source Name for the OLAP Metadata Catalog that you are creating.

Oracle: If you are using OCI direct connection, be sure to prefix the Net Service Name or the Schema Name with “Oracle” by typing `Oracle:` and then the data source name.

Note: You must create the database in which you will store the OLAP Metadata Catalog, assign the appropriate user permissions, and configure the ODBC connection before you can create the catalog.

3. In the **Code Page** drop-down list box, select the code page that corresponds to the code page of the relational database that will be used as the OLAP Metadata Catalog.

The code page is a portion of the *locale* which identifies the local language and cultural conventions such as the formatting of currency and dates and the sort order of data.

4. In the **User Name** drop-down list, select or type the user name to which you have assigned permission to access the database in which the OLAP Metadata Catalog will be stored.
5. In the **Password** text box, type the password for the user name to which you have assigned permission to access the database in which the OLAP Metadata Catalog will be stored.
6. If you want the **OLAP Metadata Catalog Setup** dialog box to be displayed automatically each time you start Integration Services Console, leave the **Show this dialog at Startup** check box selected.

If you do not select the **Show this dialog at Startup** check box, you can access the **OLAP Metadata Catalog Setup** dialog box from the console by selecting **Tools > Create Catalog**.

If an OLAP model or metaoutline is open when you select **Tools > Create Catalog**, you are prompted to disconnect from the current catalog (**Connections > OLAP Metadata Catalog > Disconnect**). Note that disconnecting from an OLAP Metadata Catalog causes the open OLAP model or metaoutline to close.

7. Click **Create**.

Integration Services Console creates the OLAP Metadata Catalog automatically.

Note: If you are migrating from a previous release of DB2 OLAP Integration Server, the OLAP Metadata Catalog auto-creation process upgrades the existing OLAP Metadata Catalog.

Creating the OLAP Metadata Catalog manually

If you choose to create the OLAP Metadata Catalog manually, you must run SQL scripts by using the same utility program that you normally use to create tables.

The scripts that you use to create the tables for a non-Unicode OLAP Metadata Catalog are named:

```
oc_create_database_name.sql
```

The scripts that you use to create the tables for a Unicode OLAP Metadata Catalog are named:

```
oc_create_database_name_unicode.sql
```

These scripts are listed in Table 21 on page 83, along with the utility programs with which they have been tested.

Note: IBM strongly recommends that you use the automatic installation process to create the OLAP Metadata Catalog (see “Creating or upgrading the OLAP Metadata Catalog automatically” on page 80).

To create tables for the OLAP Metadata Catalog database manually:

1. Start the utility program.
2. Connect to the database that you created for the OLAP Metadata Catalog.
3. Open the appropriate SQL script file in the `ocscript` directory.
4. Run the SQL script to build the tables.

On SQL Server, you receive a message that you did not create data or rows. This message is normal because you created only tables and columns.

5. Verify that you have created the OLAP Metadata Catalog tables.

For example, type a command such as:

```
SELECT * FROM JOIN_HINTS
```

Alternatively, you can start the applicable RDBMS utility program and verify that the OLAP Metadata Catalog has the new tables.

6. Close the utility program.

Upgrading the OLAP Metadata Catalog manually

If you have OLAP Metadata Catalogs from a release earlier than Version 8.2 and you choose not to use the automatic creation-upgrade process, you must upgrade the existing OLAP Metadata Catalogs manually. You cannot use the existing catalogs with Version 8.2 until they are upgraded.

Note: If you use Integration Services Console to create an OLAP Metadata Catalog automatically, the system upgrades existing OLAP Metadata Catalogs and you do not need to upgrade them manually (see “Creating or upgrading the OLAP Metadata Catalog automatically” on page 80). IBM recommends that you use the automatic process to create the OLAP Metadata Catalog.

You manually upgrade the tables of an OLAP Metadata Catalog by running the SQL scripts that use the same database utility program that you typically use to create tables. The SQL scripts to upgrade tables for the OLAP Metadata Catalog are in the `ocscript` directory where you installed DB2 OLAP Integration Server. The upgrade scripts are named `oc_upgrade*_database_name.sql` and are listed in Table 21 on page 83, along with the utility programs with which they have been tested.

Note: If you manually rebuild an OLAP Metadata Catalog, you must drop (delete) the catalog tables by using `oc_drop*_database_name.sql`. Then you create the OLAP Metadata Catalog tables by using `oc_create*_database_name.sql`. If you choose to rebuild a catalog manually, it is not necessary to run any upgrade scripts.

CAUTION:

If you drop (delete) an OLAP Metadata Catalog, you also delete the OLAP models and metaoutlines that it contains.

The following topics discuss various aspects of upgrading OLAP Metadata Catalogs manually:

- “SQL Scripts used to create and upgrade tables” on page 83
- “Upgrading tables in the OLAP Metadata Catalog” on page 83

SQL Scripts used to create and upgrade tables

The SQL scripts to create tables for the OLAP Metadata Catalog (listed in Table 21) are located in the `ocscript` directory where you installed Integration Services.

DB2 OLAP Integration Server provides SQL scripts for each supported RDBMS:

- `oc_create_database_name.sql` to build tables
- `oc_drop_database_name.sql` to drop tables
- `oc_upgrade651_database_name.sql` to upgrade tables from Version 8.1 (Hyperion 6.5) to Version 8.1, Fix Pack 1 (Hyperion 6.5.1). Tables created in Version 8.1, Fix Pack 2 and later do not need to be upgraded.

Note: If you need to rebuild an OLAP Metadata Catalog, you must drop (delete) the catalog tables before you build them. Remember that if you drop an OLAP Metadata Catalog, you also delete the OLAP models and metaoutlines that it contains.

The utilities listed in Table 21 have been tested to work with the SQL scripts.

Table 21. SQL Scripts for Creating an OLAP Metadata Catalog

Database	SQL Script	Utility Program
DB2	<code>oc_create_db2.sql</code> <code>oc_drop_db2.sql</code> <code>oc_upgrade651_db2.sql</code>	DB2 Command Center or >DB2 -tvf
Informix	<code>oc_create_informix.sql</code> <code>oc_drop_informix.sql</code> <code>oc_upgrade651_informix.sql</code>	DBAccess
Oracle	<code>oc_create_oracle.sql</code> <code>oc_drop_oracle.sql</code> <code>oc_upgrade651_oracle.sql</code> <code>oc_create_oracle_unicode.sql</code> <code>oc_create_oracle9i_unicode.sql</code>	SQL*Plus
MS SQL Server	<code>oc_create_sqlsrv.sql</code> <code>oc_drop_sqlsrv.sql</code> <code>oc_upgrade651_sqlsrv.sql</code>	Query Analyzer (MS SQL Server 7.0 and 2000)
Sybase	<code>oc_create_sybase.sql</code> <code>oc_drop_sybase.sql</code> <code>oc_upgrade651_sybase.sql</code>	ISQL
Teradata	<code>oc_create_teradata.sql</code> <code>oc_drop_teradata.sql</code> <code>oc_upgrade651_teradata.sql</code>	BTEQ

CAUTION:

After you update an OLAP Metadata Catalog, you cannot roll back to a previous version. The new version of OLAP Metadata Catalog is not compatible with previous releases of Integration Services. In addition, do not attempt to use the Version 8.2 catalog with previous releases of DB2 OLAP Integration Server software. Such an attempt can result in corrupted OLAP Metadata Catalog data.

Upgrading tables in the OLAP Metadata Catalog

To upgrade an OLAP Metadata Catalog manually, you upgrade the tables of the OLAP Metadata Catalog database.

To upgrade tables for the OLAP Metadata Catalog database:

1. Start the utility program.

2. Connect to the database that you created for the OLAP Metadata Catalog as the user who created the original OLAP Metadata Catalog tables.
3. Open the appropriate SQL script file in the `ocscript` directory.
See “SQL Scripts used to create and upgrade tables” on page 83.
4. Run the SQL script (or scripts, if applicable) to upgrade the tables.
On SQL Server, you receive a message that you did not create data or rows. This message is normal because you created only tables and columns.
5. Verify that the new tables have been added to the existing OLAP Metadata Catalog.
You can verify this by entering a command, for example:

```
SELECT * FROM OM_DESCRIPTION
```


Additionally, you can start the applicable RDBMS utility program and verify that the OLAP Metadata Catalog has the new tables.
6. Close the utility program.

Mapping OLAP Metadata Catalogs

After you create an OLAP Metadata Catalog manually, you must map the catalog to a supported ODBC driver.

For information about supported ODBC drivers, see “Supported ODBC drivers and relational data sources” on page 14. For information about configuring OLAP Metadata Catalog as an ODBC data source, see Chapter 8, “Configuring relational data sources for DB2 OLAP Integration Server,” on page 63.

Note: If you attempt to access an SQL Server database with the Microsoft native ODBC driver and you do not have access permission, SQL Server connects you to the default database without notifying you. To use Integration Services on Sybase, you must have execute permission for `sp-fkeys` in `Sybsystemprocs`.

Deleting OLAP Metadata Catalogs

You can delete an existing OLAP Metadata Catalog at any time.

CAUTION:

Remember that if you delete an OLAP Metadata Catalog, you also delete the OLAP models and metaoutlines that it contains.

To delete an OLAP Metadata Catalog:

1. From either the OLAP Model or OLAP Metaoutline main window, select **Tools > Create Catalog** to display the **OLAP Metadata Catalog Setup** dialog box.
2. From the **Server Name** drop-down list, select the appropriate DB2 OLAP Integration Server computer.
3. From the **Catalog ODBC DSN** drop-down list, select the Data Source Name for the OLAP Metadata Catalog that you want to delete.
4. In the **User Name** drop-down list, select or type your user name.
5. In the **Password** text box, type your password.
6. Click **Yes** to the confirmation prompt to delete the catalog.

If the OLAP Metadata Catalog that you are attempting to delete is currently in use, you are prompted to disconnect from the current catalog (**Connections > OLAP Metadata Catalog > Disconnect**). You must disconnect before you can delete the catalog.

Note: Disconnecting from an OLAP Metadata Catalog causes the open OLAP model or metaoutline to close.

Chapter 10. Setting up the sample application for DB2 OLAP Integration Server

DB2 OLAP Integration Server provides a sample application that consists of the following features:

- A database for a fictitious company called The Beverage Company (TBC)
- A sample OLAP model and metaoutline created from the TBC database
- An OLAP model and metaoutline enabled for Hybrid Analysis
- A special OLAP Metadata Catalog to store the sample OLAP models and metaoutlines and any additional OLAP models and metaoutlines that you create using the sample data

Preparing the sample application consists of three tasks:

- Setting up the standard sample application
- **Optional:** Setting up the Hybrid Analysis sample application
- **Optional:** Setting up the Unicode sample application

You can set up the standard sample application automatically using Integration Services Console (see “Setting up the standard sample application automatically” on page 89), or you can set up the standard sample application manually (“Setting up the standard sample application manually” on page 90). Using either method, setting up the standard sample application involves two main tasks:

- You create two relational databases: one for the data in the TBC sample application and one for the OLAP Metadata Catalog database (TBC_MD).

Note: Creating the relational databases is a manual task that you must perform prior to setting up the standard sample application, regardless of whether you choose the automatic or manual method.

- You then create tables in each of these databases and load data into them, using scripts provided with Integration Services. You can create tables either automatically, through the use of Integration Services Console, or manually.

When you finish setting up the standard sample application, the OLAP Metadata Catalog database (TBC_MD) will contain an OLAP model (TBC Model) and a metaoutline (TBC Metaoutline) based on data in the sample TBC relational data source.

You create the Hybrid Analysis sample application manually by completing the tasks in “Setting up the Hybrid Analysis sample application” on page 96. Upon completion of these tasks, the sample TBC relational data source will contain Hybrid Analysis data. The OLAP Metadata Catalog will contain an OLAP model and metaoutline enabled for Hybrid Analysis (HA TBC Model and HA TBC Metaoutline, respectively).

CAUTION:

If you have installed the sample application from a previous release of DB2 OLAP Integration Server, you should back up and preserve your existing sample database, OLAP Metadata Catalog, and the OLAP models and metaoutlines that OLAP Metadata Catalog contains. You can then upgrade your existing catalog to be compatible with the current release of the software (see “Upgrading the OLAP Metadata Catalog manually” on page 82). You cannot, however, store newly created OLAP models and metaoutlines in your previous catalog.

This chapter tells you what to consider before and after you set up the standard sample application, whether automatically or manually. For manual setup, the topics in this chapter detail which scripts and batch files you must run to create and load the TBC database and the TBC_MD OLAP Metadata Catalog.

The procedures in this chapter assume that you know how to create a database using a relational database management system (RDBMS). For manual setup, the procedures assume that you know how to create tables and load them with data running SQL scripts, batch files, and shell scripts. For information on these topics, see the documentation for the RDBMS that you are using.

This chapter contains the following topics:

- “Preparing to set up the standard sample application” on page 88
- “Setting up the standard sample application automatically” on page 89
- “Setting up the standard sample application manually” on page 90
- “Setting up the TBC relational data source” on page 91
- “Setting up the TBC_MD OLAP Metadata Catalog” on page 93
- “Setting up the Hybrid Analysis sample application” on page 96
- “Setting up the Unicode sample application” on page 104
- “After setting up the sample application” on page 106

Preparing to set up the standard sample application

Before you set up the standard sample application, you must install the database client software for a supported RDBMS. When setting up the standard sample application, you must use a computer on which the server component of DB2 OLAP Integration Server and the necessary database client software are both installed. For more information, see “About data source configuration” on page 63.

For manual setup of the standard sample application, the standard sample application scripts do not have to be installed on the same computer that you are using, but you must be able to access the script files.

Note: IBM strongly recommends that you use the automatic installation process to install the standard sample applications (see “Setting up the standard sample application automatically” on page 89).

In addition to the sample application scripts, the standard sample application requires the database client software for the RDBMS that you are using. The database client software is required to run the standard sample application scripts.

If you are using Oracle OCI direct connection, verify that the database client software for the RDBMS is installed. If you choose to set up the standard sample application manually, refer to Table 22 on page 89 and also verify that the appropriate client utility program is installed.

Table 22. Required Utilities for Setting Up the Standard Sample Application

Database	Utility Programs
DB2 UDB	DB2 Command Window or >DB2 -tvf
Informix	DBAccess
MS SQL Server	Query Analyzer (MS SQL Server 7.0 and 2000)
Oracle	SQL*Plus
Sybase	<ul style="list-style-type: none"> • ISQL • BCP command line utility
Teradata	BTEQ

The remainder of this chapter assumes that you have installed the required database client software.

Setting up the standard sample application automatically

After you have completed the Integration Services installation process, you can choose to install either or both the OLAP Metadata Catalog and the standard sample application. The system first displays the OLAP Metadata Catalog Setup dialog box to enable automatic setup of the catalog (see “Creating or upgrading the OLAP Metadata Catalog automatically” on page 80). Whether or not you choose to create the OLAP Metadata Catalog automatically, you can choose to create the standard sample application automatically, after you finish creating the OLAP Metadata Catalog. You can create the standard sample application quickly and easily using the automatic feature. You are encouraged to use this convenient feature, but if you choose manual setup, see “Setting up the standard sample application manually” on page 90 for detailed procedures.

Hybrid Analysis: If you plan to set up the sample application for Hybrid Analysis, you must follow the procedures in “Setting up the Hybrid Analysis sample application” on page 96.

Note: ODBC connections must be configured before you begin to set up the standard sample application. If you have not yet configured ODBC connections, see Chapter 8, “Configuring relational data sources for DB2 OLAP Integration Server,” on page 63 for information on setting up ODBC data sources, including an OLAP Metadata Catalog and relational data sources.

To set up the standard sample application automatically:

1. Select **Tools > Create Sample** to access the **Sample Application Setup** dialog box.
2. In the **Sample ODBC DSN** text box, select or type the data source name for creating or dropping (deleting) tables in the source database.

Oracle OCI: If you are using Oracle OCI direct connection, be sure to prefix the Net Service Name or the Schema Name with “Oracle” by typing Oracle: and then the data source name.

The User Name text box displays the name, TBC (lowercase tbc for Informix). You must use this name to create the TBC standard sample application database.

3. In the **Password** text box, type the password for the user to which you have assigned privileges to access the TBC database for the standard sample application.
4. Click **Create** to create the standard sample application.

DB2 OLAP Integration Server creates the necessary tables, loads the TBC sample data into the tables, and loads the sample OLAP model and metaoutline data into the OLAP Metadata Catalog.

If DB2 OLAP Integration Server detects that you are not currently connected to the OLAP Metadata Catalog in which to load the sample OLAP model and metaoutline data, the **Login** dialog box is displayed so that you can enter the information necessary to connect. After you close the **Login** dialog box and are returned to the **Sample Application Setup** dialog box, click **Create** again to initiate setup of the standard sample application.

If, after clicking **Create**, DB2 OLAP Integration Server detects either an existing sample data source or an existing sample OLAP model and metaoutline in the OLAP Metadata Catalog, you are prompted with the message that tables already exist in the database. Click **OK** to return to the **Sample Application Setup** dialog box to reenter information, or click **Delete** to delete the existing standard sample application data. If you click **Delete**, your existing OLAP Metadata Catalog sample OLAP models and metaoutlines will be deleted and replaced with the sample OLAP model and metaoutline for the current release.

After you have completed creation of the standard sample application, you are ready to begin working with Integration Services Console to create your own sample OLAP models and metaoutlines (see “Starting Integration Services Console” on page 57).

Setting up the standard sample application manually

To set up the standard sample application manually, you begin by creating two databases in a supported RDBMS: TBC (the sample database) and TBC_MD (an OLAP Metadata Catalog). You then create tables in each of these databases and load data into them, using scripts provided with DB2 OLAP Integration Server. The names and directory locations of these scripts are detailed in the topics that follow.

Note: IBM strongly recommends that you use the automatic installation process to install the standard sample applications (see “Setting up the standard sample application automatically” on page 89).

After you create the TBC and TBC_MD databases in a supported RDBMS, you use several scripts and files to create the standard sample application:

- A script for creating the TBC sample database tables
This script is located in the `samples\tbcdbase` directory.
- A script for loading the TBC sample database data
This script is located in the `samples\tbcdbase` directory.
- A script for creating the TBC_MD OLAP Metadata Catalog database tables
This script is located in the `ocscript` directory.
- XML files for importing a sample OLAP model and metaoutline in the TBC_MD OLAP Metadata Catalog database
These XML files are located in the `samples\tbcmodel` directory.

The scripts for creating the standard sample application and OLAP Metadata Catalog are installed with the server software. Different versions of the scripts are provided for each of the supported RDBMSs.

Note: Some of the standard sample application scripts require slightly different procedures, depending on the RDBMS that you are using.

Setting up the TBC relational data source

The relational data source for the standard sample application is TBC.

Note: To create a database, you must have database administrator or similar access privileges required by the RDBMS that you are using.

To set up TBC relational data source:

1. Create the TBC database using an RDBMS.
For more information, see “Creating the TBC database.”
2. Create tables for the TBC database by running SQL scripts.
For instructions, see “Creating tables for the TBC database.”
3. Load data into the TBC database by running SQL scripts.
For instructions, see “Loading data into the TBC tables” on page 92.

Creating the TBC database

Create the TBC database in the same way that you create any database using an RDBMS:

- Create a database device or tablespace named TBC.
- Allocate 20 MB for storage.
- Create a user TBC who can drop and create tables.
- Grant user privileges or permissions to create and drop tables.

CAUTION:

The user TBC must create the tables for the TBC database, or portions of the standard sample application will not work.

Creating tables for the TBC database

Create tables for the TBC database with the same utility program that you typically use to create tables by running SQL scripts.

The standard sample application SQL scripts needed to create tables for the TBC database are in the `samples\tbcdbase` directory where you installed DB2 OLAP Integration Server.

The utilities listed in Table 23 have been tested to work with the SQL scripts:

Table 23. Tested Utilities for Creating TBC Tables

Database	SQL Script	Utility Program
DB2	tbc_create_db2.sql tbc_drop_db2.sql	DB2 Command Window or >DB2 -tvf
Informix	tbc_create_informix.sql tbc_drop_informix.sql	DBAccess

Table 23. Tested Utilities for Creating TBC Tables (continued)

Database	SQL Script	Utility Program
MS SQL Server	tbc_create_sqlsrv.sql tbc_drop_sqlsrv.sql	Query Analyzer (MS SQL Server 7.0 and 2000)
Oracle	tbc_create_oracle.sql tbc_drop_oracle.sql	<ul style="list-style-type: none"> • SQL*Plus • SQL*Loader command line utility
Sybase	tbc_create_sybase.sql tbc_drop_sybase.sql	ISQL
Teradata	tbc_create_teradata.sql tbc_drop_teradata.sql	BTEQ

Each RDBMS has two scripts. The first script builds tables and the other drops tables.

CAUTION:

You must create the tables for the TBC database as user TBC, or portions of the standard sample application will not work.

To create tables for the TBC database:

1. Start the utility program.
2. Connect to the TBC database as user TBC.

Note: Use lowercase tbc for Informix; use uppercase TBC for Sybase and SQL Server.

3. In the samples\tbcdbase directory, open the appropriate SQL script file.
4. Run the SQL script.

MS-SQL Server: You should receive a message that you did not create data or rows. This message is normal because you created only tables and columns.

Informix: If you run the Informix SQL script (tbc_create_informix.sql) more than once, you must first run tbc_drop_informix.sql to drop tables before you build them again.

5. Verify that you have created the TBC tables; for example, type

```
SELECT * FROM PRODUCTDIM
```

Alternatively, you can start the RDBMS and verify that the TBC database has the new tables.

6. Close the utility program.

Loading data into the TBC tables

Load data into the TBC tables by running an SQL script using the same utility program that you typically use to load tables by running SQL scripts.

The utilities listed in Table 24 have been tested to work with SQL scripts.

Table 24. Tested Utilities for Loading Data into TBC Tables

Database	SQL Script	Utility Program
DB2	sampldata.sql	DB2 Command Window or >DB2 -tvf
Informix	sampldata.sql	DBAccess
MS SQL Server	sampldata.sql	Query Analyzer (MS SQL Server 7.0 and 2000)
Oracle	sampldata.sql	SQL*Plus

Table 24. Tested Utilities for Loading Data into TBC Tables (continued)

Database	SQL Script	Utility Program
Sybase	sampledata.sql	ISQL
Teradata	sampledata.sql	BTEQ

To load data into the TBC tables:

1. From the command line, move to the `samples\tbcdbase` directory where you installed Integration Server.
2. Start the utility program.
3. Connect to the TBC database as user TBC.
4. In the `sample\tbcdbase` directory, open the `sampledata.sql` script file.
5. Run the SQL script using your RDBMS tool.
6. Verify that you have loaded data into the TBC tables; for example, type

```
SELECT * FROM TBC.PRODUCT
```

Alternatively, you can start the RDBMS and execute a query.
7. Close the utility program.

Setting up the TBC_MD OLAP Metadata Catalog

The OLAP Metadata Catalog for the standard sample application is TBC_MD. For more information on the OLAP Metadata Catalog, see Chapter 9, “Creating, upgrading, and deleting OLAP Metadata Catalogs,” on page 79.

To set up the TBC_MD OLAP Metadata Catalog:

1. Create a TBC_MD database using an RDBMS.
For more information, see “Creating the TBC_MD database.”
2. Create tables for the TBC_MD database by running SQL scripts.
For instructions, see “Creating Tables for the TBC_MD Database” on page 94.
3. Load data into the TBC_MD database using the XML import utility.
For instructions, see “Using the XML Import utility to load data into the TBC_MD tables” on page 95.

Note: If you have a previous release of DB2 OLAP Integration Server and are upgrading your OLAP Metadata Catalog, you cannot roll back to the previous version. The new version of the OLAP Metadata Catalog is not compatible with earlier releases of DB2 OLAP Integration Server.

Creating the TBC_MD database

Create the TBC_MD database in the same way that you create any database using an RDBMS:

- Create a database device or tablespace named TBC_MD.
- Allocate 20 MB for storage.
- Create a user TBC who can drop and create tables.
- Grant user privileges or permissions.

CAUTION:

You must create the tables for the TBC_MD database as user TBC or portions of the standard sample application will not work.

Creating Tables for the TBC_MD Database

Create tables for the TBC_MD database with the same utility program that you typically use.

The standard sample application SQL scripts used to create tables for the TBC_MD database are in the ocscript directory where you installed DB2 OLAP Integration Server.

The SQL scripts in the ocscript directory are the same scripts that you use to create any OLAP Metadata Catalog. For information on OLAP Metadata Catalogs, see Chapter 9, "Creating, upgrading, and deleting OLAP Metadata Catalogs," on page 79.

The utilities listed in Table 25 have been tested to work with the SQL scripts:

Table 25. Tested Utilities for Creating TBC_MD Tables

Database	SQL Script	Utility Program
DB2	oc_create_db2.sql oc_drop_db2.sql oc_upgrade651_db2.sql	DB2 Command Center or >DB2 -tvf
Informix	oc_create_informix.sql oc_drop_informix.sql oc_upgrade651_informix.sql	DBAccess
MS SQL Server	oc_create_sqlsrv.sql oc_drop_sqlsrv.sql oc_upgrade651_sqlsrv.sql	Query Analyzer (MS SQL Server 7.0 and 2000)
Oracle	oc_create_oracle.sql oc_drop_oracle.sql oc_upgrade651_oracle.sql oc_create_oracle_unicode.sql oc_create_oracle9i_unicode.sql	SQL*Plus
Sybase	oc_create_sybase.sql oc_drop_sybase.sql oc_upgrade651_sybase.sql	ISQL
Teradata	oc_create_teradata.sql oc_drop_teradata.sql oc_upgrade651_teradata.sql	BTEQ

DB2 OLAP Integration Server provides the following SQL scripts for each RDBMS (with the exception of Teradata):

- oc_create_database_name.sql to build tables
- oc_drop_database_name.sql to drop tables
- oc_upgrade651_database_name.sql to upgrade tables from Version 8.1 (Hyperion 6.5) to Version 8.1, Fix Packs 1 and 2 (Hyperion 6.5.1)

If you need to rebuild tables, you must first drop the tables before you build them again.

CAUTION:

You must create the tables for the TBC_MD database as user TBC or portions of the standard sample application will not work.

To create tables for the TBC_MD database:

1. Start the utility program.

2. Connect to the TBC_MD database as user TBC.
3. In the ocscrip directory, open the appropriate SQL script file.
4. Run the SQL script.

MS-SQL Server: You receive a message that you did not create data or rows. This message is normal because you created only tables and columns.

5. Verify that you have created the TBC_MD tables; for example, type the following command:

```
SELECT * FROM TBC.MS_INFO
```

 Alternatively, start the RDBMS and verify that the TBC_MD database has the new tables.
6. Close the utility program.

Using the XML Import utility to load data into the TBC_MD tables

Use the XML Import utility to load OLAP model and metaoutline data into the TBC_MD tables.

The XML files listed in Table 26 have been tested to work with their respective RDBMS. These files are located in the `samples\tbcmodel` directory.

Table 26. XML Files for Loading Data into TBC_MD Tables

Database	XML File in <code>sample\tbcmodel</code>
<ul style="list-style-type: none"> • DB2 • Oracle • MS SQL Server • Sybase • Teradata 	<pre>sample_model.xml sample_metaoutline.xml</pre>
Informix	<pre>sample_model_informix.xml sample_metaoutline.xml</pre>

Using XML Import to load data into TBC_MD

To complete the procedure below, you must start Integration Services Console and connect to Integration Server.

To start Integration Services Console and connect to Integration Server:

1. On the Windows desktop, click **Start > Programs > IBM DB2 OLAP Server 8.2 > IBM DB2 OLAP Integration Server > Integration Server**.
2. On the Windows desktop, click **Start > Programs > IBM DB2 OLAP Server 8.2 > IBM DB2 OLAP Integration Server > Desktop**.
3. If necessary, click **Close** to clear the **OLAP Metadata Catalog Setup** dialog box. Integration Services Console automatically displays the Login dialog box.
 - a. In the **Server** text box, under DB2 OLAP Integration Server, select or type the name of a server computer; for example, `cypress`.

If the port number on which DB2 OLAP Integration Server communicates with the console has been set to a value other than the default, you must type the server name or IP address and type the non-standard port number, separated by a colon; for example:

```
cypress:3390
```

- b. In the **OLAP Metadata Catalog** text box, select or type the name of the OLAP Metadata Catalog, TBC_MD, for the sample TBC (The Beverage Company) database.

Oracle OCI: If you are using Oracle OCI direct connection, be sure to prefix the Net Service Name or the Schema Name with “Oracle” by typing Oracle: and then the data source name.

- c. In the **User Name** and **Password** text boxes, type the user name and password for the standard sample application user, TBC, and click **OK**.

To load sample OLAP model data into the TBC_MD tables using XML Import:

1. In Integration Services Console, select **File > XML Import/Export**.
2. Select the **Import** tab.
3. In the **XML File Path** text box, enter the file path for the sample OLAP model or click the **Open XML File** button and navigate to the following file:

is\Samples\tbcmodel

4. Select the XML file to import for the sample OLAP model data, sample_model.xml.

Informix: Select the XML file to import for the sample OLAP model for Informix, sample_model_informix.xml.

5. Click **OK**.

Integration Services Console displays the XML Import/Export dialog box with the XML file path and name in the XML File Path text box. The name of the sample OLAP model is displayed in the **OLAP Model Name** text box.

6. Click the **Import to Catalog** button to load the selected XML file into the sample OLAP Metadata Catalog.

To load sample OLAP metaoutline data into the TBC_MD tables using XML Import:

1. In Integration Services Console, select **File > XML Import/Export**.
2. Select the **Import** tab.
3. In the **XML File Path** text box, enter the file path for the sample OLAP metaoutline or click the **Open XML File** button and navigate to the following file:

is\Samples\tbcmodel

4. Select the XML file to import for the sample OLAP metaoutline data, sample_metaoutline.xml.

5. Click **OK**.

Integration Services Console displays the XML Import/Export dialog box with the XML file path and name in the XML File Path text box, the name of the sample OLAP metaoutline in the OLAP Metaoutline Name text box, and the name of the OLAP model upon which this metaoutline is based in the Based on OLAP Model text box.

6. Click the **Import to Catalog** button to load the selected XML file into the sample OLAP Metadata Catalog.

Setting up the Hybrid Analysis sample application

Before setting up the Hybrid Analysis sample application, you must set up the standard sample application (see “Setting up the standard sample application automatically” on page 89 or “Setting up the standard sample application manually” on page 90).

Setting up the Hybrid Analysis sample application is a manual process. Follow the instructions given in this section for your particular platform.

Note: The Hybrid Analysis application scripts require slightly different procedures, depending on the RDBMS that you are using.

The utilities listed in 27 are required for running the various batch files and SQL scripts used in creating the Hybrid Analysis sample application.

Table 27. Required Utilities for Setting Up the Hybrid Analysis Sample Application

Database	Required Utilities
DB2	DB2CMD command line interface
Oracle	<ul style="list-style-type: none"> • SQL*Plus • SQL*Loader command line utility
SQL Server	<ul style="list-style-type: none"> • ISQL • BCP command line utility
Teradata	<ul style="list-style-type: none"> • BTEQ • FastLoad

Setting up the Hybrid Analysis sample application on DB2 UDB

You use several files to create the Hybrid Analysis sample application on DB2 Universal Database:

- A batch file to send commands to the DB2CMD command line interface:
install_db2.bat
- A data load script for the TBC sample database:
ha_create_db2.sql
- An XML file that you use to import a sample OLAP model into the TBC_MD OLAP Metadata Catalog database:
HA_TBC_Model.xml
- An XML file that you use to import a sample metaoutline into the TBC_MD OLAP Metadata Catalog database:
HA_TBC_Metaoutline.xml
- A text file containing sample data:
ha_sampledata.txt

Setting up the Hybrid Analysis sample application on DB2 UDB consists of performing the following tasks:

- Running the batch file install_db2.bat
- Importing the sample OLAP model and metaoutline

Note: Before setting up the Hybrid Analysis Sample Application on DB2 UDB, verify that the DB2CMD command line interface is installed.

Running the batch file on DB2 UDB

The first task in setting up the Hybrid Analysis sample application on DB2 UDB consists of running the install_db2.bat file. This batch file creates tables in the sample database, loads metadata into the dimension tables, and loads sample data into the fact table.

Run the install_db2.bat file located in:

`\is\Samples\tbcdbase`

This batch file sends SQL commands to the DB2CMD command line interface.

Pass the following three parameters to install_db2.bat:

- The user ID that you use to connect with DB2
- Your password
- The DB2 UDB database name

The install_db2.bat file runs the DB2CMD command line interface, which executes the ha_create_db2.sql load script file. This script file performs the following tasks:

- Deletes any existing tables in the sample TBC database
- Creates a new set of tables in the TBC database
- Inserts records into the dimension tables
- Creates the ha_results_createtables.txt file
- Loads the sample data into the fact table

The ha_results_createtables.txt file contains the results of the table-creating and loading operations.

Note: After the member and data load is finished, close the DB2 UDB command window.

Importing the sample OLAP model and metaoutline

The final task in setting up the Hybrid Analysis sample application on DB2 UDB consists of importing the two XML files for the sample OLAP model and metaoutline:

To import the sample OLAP model:

1. Start Integration Services Console.
2. Select **File > XML Import/Export**.
The XML Import/Export window is displayed.
3. Select the **Import** tab.
4. In the **XML File Path** text box, enter the file path for the sample OLAP model or click the **Open XML File** button and navigate to the following file:
`\is\Samples\tbcmodel\HA_TBC_Model.xml`
5. Click **Open XML File**.

To import the sample metaoutline:

1. Select **File > XML Import/Export**.
The XML Import/Export window is displayed.
2. Select the **Import** tab.
3. In the **XML File Path** text box, enter the file path for the sample OLAP metaoutline or click the **Open XML File** button and navigate to the following file:
`\is\Samples\tbcmodel\HA_TBC_Metaoutline.xml`
4. Click **Open XML File**.

Setting up the Hybrid Analysis sample application on Oracle

You use several files to create the Hybrid Analysis sample application on Oracle:

- A batch file to send commands to the SQLPLUS command line interface:
install_oracle.bat
- A data load script for the TBC sample database:
ha_create_oracle.sql
- An Oracle control file to load data into the fact table:
hasales.ctl
- An XML file that you use to import a sample OLAP model into the TBC_MD OLAP Metadata Catalog database:
HA_TBC_Model.xml
- An XML file you use to import a sample metaoutline into the TBC_MD OLAP Metadata Catalog database:
HA_TBC_Metaoutline.xml
- A text file containing sample data:
ha_sampledata.txt

Setting up the Hybrid Analysis sample application on Oracle consists of performing the following tasks:

- Running the batch file install_oracle.bat
- Importing the sample OLAP model and metaoutline

Note: Before setting up the Hybrid Analysis Sample Application on Oracle, verify that SQL*PLUS and SQLLDR command line interfaces are installed.

Running the batch file

The first task in setting up the Hybrid Analysis sample application on Oracle consists of running the install_oracle.bat file. This batch file creates tables in the sample database, loads metadata into the dimension tables, and loads sample data into the fact table.

Run the install_oracle.bat file located in:

```
\is\Samples\tbcdbase
```

This batch file sends SQL commands to the SQL*PLUS command line interface.

Pass the following three parameters to install_oracle.bat :

- The user ID that you use to connect with Oracle
- Your password
- The Oracle service name as defined in the TNSNAMES.ORA file

Metadata load script: The install_oracle.bat file runs the SQLPLUS command line interface, which executes the ha_create_oracle.sql metadata load script file. This script file performs the following tasks:

- Deletes any existing tables in the sample TBC database
- Creates a new set of tables in the TBC database
- Inserts records into the dimension tables
- Creates the ha_results_createtables.txt file

The ha_results_createtables.txt file contains the results of the table-creating operation.

Sample data load script: After loading the metadata into the dimension tables, the `install_oracle.bat` file runs the SQLLDR command line interface, which performs the following tasks:

- Loads the sample data into the fact table
- Creates the `ha_results_loadfact.txt` file

The `ha_results_loadfact.txt` file contains the results of the data-loading operation.

Importing the sample OLAP model and metaoutline

The final task in setting up the Hybrid Analysis sample application on Oracle consists of importing the two XML files for the sample OLAP model and metaoutline:

To import the sample OLAP model:

1. Start Integration Services Console.
2. Select **File > XML Import/Export**.
The XML Import/Export window is displayed.
3. Select the **Import** tab.
4. In the **XML File Path** text box, enter the file path for the sample OLAP model or click the **Open XML File** button and navigate to the following file:
`\is\Samples\tbcmodel\HA_TBC_Model.xml`
5. Click **Open XML File**.

To import the sample metaoutline:

1. Select **File > XML Import/Export**.
The XML Import/Export window is displayed.
2. Select the **Import** tab.
3. In the **XML File Path** text box, enter the file path for the sample OLAP metaoutline or click the **Open XML File** button and navigate to the following file:
`\is\Samples\tbcmodel\HA_TBC_Metaoutline.xml`
4. Click **Open XML File**.

Setting up the Hybrid Analysis sample application on SQL Server

Several files are used to create the Hybrid Analysis sample application on SQL Server:

- A batch file to send commands to MS Interface SQL (ISQL):
`install_sqlsrv.bat`
- A data load script for the TBC sample database:
`ha_create_sqlsrv.sql`
- An XML file that you use to import a sample OLAP model into the TBC_MD OLAP Metadata Catalog database:
`HA_TBC_Model.xml`
- An XML file that you use to import a sample metaoutline into the TBC_MD OLAP Metadata Catalog database:
`HA_TBC_Metaoutline.xml`
- A text file containing sample data:
`ha_sampledata.txt`

Setting up the Hybrid Analysis sample application on SQL Server consists of performing the following tasks:

- Running the `install_sqlsrv.bat` batch file
- Importing the sample OLAP model and metaoutline

Note: Before setting up the Hybrid Analysis Sample Application on SQL Server, verify that you have ISQL and BCP (Bulk Copy) installed. These utilities were most likely included when you installed SQL Server Client.

Running the batch file

To set up the Hybrid Analysis sample application on SQL Server, you first run the `install_sqlsrv.bat` file. This batch file creates tables in the sample database, loads metadata into the dimension tables, and loads sample data into the fact table. Run the `install_sqlsrv.bat` file located in:

```
\\is\Samples\tbcdbase
```

This file sends commands to the Interface SQL (ISQL) command line interface. These commands install the sample application to the sample database TBC.

Pass the following three parameters to `install_sqlsrv.bat`:

- The user ID that you use to connect with the SQL Server (TBC)
- Your password
- The SQL Server name (name of machine where the SQL Server is installed)

Metadata load script: The `install_sqlsrv.bat` file runs the ISQL utility, which executes the `ha_create_sqlsrv.sql` metadata load script file. This file performs the following tasks:

- Deletes any existing tables in the sample TBC database
- Creates a new set of tables in the TBC database
- Inserts records into the dimension tables
- Creates the `ha_results_createtables.txt` file

The `ha_results_createtables.txt` file contains the results of the table-creating operation.

Sample data load script: After loading the metadata into the dimension tables, the `install_sqlsrv.bat` file runs the BCP (Bulk Copy) utility which performs the following tasks:

- Loads the sample data into the fact table
- Creates the `ha_results_loadfact.txt` file

The `ha_results_loadfact.txt` file contains the results of the data-loading operation.

Importing the sample OLAP model and metaoutline

The final task in setting up the Hybrid Analysis sample application on SQL Server consists of importing the two XML files for the sample OLAP model and metaoutline:

To import the sample OLAP model:

1. Start Integration Services Console.
2. Select **File > XML Import/Export**.

The XML Import/Export window is displayed.

3. Select the **Import** tab.
4. In the **XML File Path** text box, enter the file path for the sample OLAP model or click the **Open XML File** button and navigate to the following file:
`\\is\Samples\tbcmodel\HA_TBC_Model.xml`
5. Click **Open XML File**.

To import the sample metaoutline:

1. Select **File > XML Import/Export**.
 The XML Import/Export window is displayed.
2. Select the **Import** tab.
3. In the **XML File Path** text box, enter the file path for the sample OLAP metaoutline or click the **Open XML File** button and navigate to the following file:
`\\is\Samples\tbcmodel\HA_TBC_Metaoutline.xml`
4. Click **Open XML File**.

Setting up the Hybrid Analysis sample application on Teradata

Several files create the Hybrid Analysis sample application on Teradata:

- A batch file to send commands to MS Interface SQL (ISQL):
`install_teradata.bat`
- A data load script for the TBC sample database:
`ha_create_teradata.sql`
- An SQL file to add constraints to the fact table after the data load has been completed:
`ha_altertable_teradata.sql`
- A FastLoad script file, which loads data to the fact table:
`ha_fastload_teradata.sql`
- An XML file that you use to import a sample OLAP model into the TBC_MD OLAP Metadata Catalog database:
`HA_TBC_Model.xml`
- An XML file that you use to import a sample metaoutline into the TBC_MD OLAP Metadata Catalog database:
`HA_TBC_Metaoutline.xml`
- A text file containing sample data:
`ha_sampledata.txt`

Setting up the Hybrid Analysis sample application on SQL Server consists of performing the following tasks:

- Running the batch file `install_teradata.bat`
- Importing the sample OLAP model and metaoutline

Note: Before setting up the Hybrid Analysis Sample Application on Teradata, verify that the BTEQ and FastLoad utilities are installed.

Running the batch file

The first task in setting up the Hybrid Analysis sample application on Teradata consists of running the `install_teradata.bat` file. This batch file creates tables in the sample database, loads metadata into the dimension tables, loads sample data into the fact table, and adds constraints to the fact table.

Run the `install_teradata.bat` file located in:

```
\is\Samples\tbcdbase
```

Note: If you installed DB2 OLAP Integration Server on Windows, the batch file is in the following directory:

```
\is\Samples\tbcdbase
```

This batch file sends SQL commands to the BTEQ command line interface.

Pass the following three required parameters to `install_teradata.bat`:

- The user ID that you use to connect with Teradata
- Your password
- The Teradata database name (DBC)

You can also pass, as an optional parameter, the database name where you want to install the sample application. (The default database is TBC.)

Metadata load script: The `install_teradata.bat` file runs the BTEQ utility, which executes the `ha_create_teradata.sql` metadata load script file. This script file performs the following tasks:

- Deletes any existing tables in the sample TBC database
- Creates a new set of tables in the TBC database
- Inserts records into the dimension tables
- Creates the `ha_results_createtables.txt` file

The `ha_results_createtables.txt` file contains the results of the table-creating operation.

Sample data load script: After loading the metadata into the dimension tables, the `install_teradata.bat` file runs the FastLoad utility, which performs the following tasks:

- Loads the sample data into the fact table using the `ha_fastload_teradata.txt` file
- Creates the `ha_results_loadfact.txt` file

The `ha_results_loadfact.txt` file contains the results of the data-loading operation.

Alter table script: After loading the sample data into the fact table, the `install_teradata.bat` file runs the BTEQ utility, which executes the `ha_altertable_teradata.sql` alter table script file. This script file performs the following tasks:

- Deletes a few tables no longer needed in the sample TBC database
- Creates indexes and places constraints on the fact table
- Creates the `ha_results_altertables.txt` file

The `ha_results_altertables.txt` file contains the results of the table-altering operation.

Importing the Sample OLAP Model and Metaoutline

The final task in setting up the Hybrid Analysis sample application on Teradata consists of importing the two XML files for the sample OLAP model and metaoutline:

To import the sample OLAP model:

1. Start Integration Services Console.
2. Select **File > XML Import/Export**.
The XML Import/Export window is displayed.
3. Select the **Import** tab.
4. In the **XML File Path** text box, enter the file path for the sample OLAP model or click the **Open XML File** button and navigate to the following file:
`\\is\Samples\tbcmodel\HA_TBC_Model.xml`
5. Click **Open XML File**.

To import the sample metaoutline:

1. Select **File > XML Import/Export**.
The XML Import/Export window is displayed.
2. Select the **Import** tab.
3. In the **XML File Path** text box, enter the file path for the sample OLAP metaoutline or click the **Open XML File** button and navigate to the following file:
`\\is\Samples\tbcmodel\HA_TBC_Metaoutline.xml`
4. Click **Open XML File**.

Setting up the Unicode sample application

Setting up the Unicode sample application is a manual process.

The utilities listed in Table 28 are required for running the various batch files and SQL scripts used in creating the Unicode sample application.

Table 28. Required Utilities for Setting Up the Unicode Sample Application

Database	Required Utilities
Oracle	<ul style="list-style-type: none">• SQL*Plus• SQL*Loader command line utility

Creating the Unicode TBC_U database

Create the Unicode TBC_U database in the same way that you create any database using an RDBMS:

- Create a database device or tablespace named TBC_U.
- Allocate 20 MB for storage.
- Create a user TBC_U who can drop and create tables.
- Grant user privileges or permissions to create and drop tables.

CAUTION:

You should create the tables for the Unicode-enabled database as user TBC_U, or portions of the Unicode sample application may not work.

Creating the Unicode TBC_MD_U OLAP Metadata Catalog

Create the Unicode TBC_MD_U OLAP Metadata Catalog by following the steps in “Setting up the TBC_MD OLAP Metadata Catalog” on page 93, using the file `oc_create_database_name_unicode.sql`.

Setting up the Unicode sample application

You use several files to create the Unicode sample application:

- A table creation and data load script for the Unicode sample database:
`tbc_create_database_name_unicode.sql`
- An XML file that you use to import a sample OLAP model into the TBC_MD_Unicode OLAP Metadata Catalog database:
`TBC_Model_Unicode.xml`
- An XML file that you use to import a sample metaoutline into the TBC_MD_Unicode OLAP Metadata Catalog database:
`TBC_Metaoutline_Unicode.xml`

Setting up the Unicode sample application consists of performing the following tasks:

- Running the SQL file `tbc_create_database_name_unicode.sql`
- Importing the Unicode sample model
- Importing the Unicode sample metaoutline

Running the SQL file

The first task in setting up the Unicode sample application consists of running the `tbc_create_database_name_unicode.sql` file using one of the utilities listed in Table 28 on page 104.

Note: Make sure you have created a Unicode-enabled database for your sample application as described in “Creating the Unicode TBC_U database” on page 104.

Run the `tbc_create_database_name_unicode.sql` file located in:
`\is\samples\tbcbase\`

The `tbc_create_database_name_unicode.sql` file performs the following tasks:

- Deletes any existing tables in the Unicode sample database
- Creates a new set of tables in the Unicode sample database
- Loads Unicode sample metadata into the dimension tables
- Loads Unicode sample data into the fact table

Importing the Unicode sample model and metaoutline

The final task in setting up the Unicode sample application consists of importing the XML file for the Unicode sample model:

To import the Unicode sample OLAP model:

1. Start Integration Services Console and log in to the Unicode catalog.
2. Select **File > XML Import/Export**.
The XML Import/Export dialog box is displayed.
3. Select the **Import** tab.
4. In the **XML File Path** text box, enter the file path for the Unicode sample OLAP model or click the **Open XML File** button and navigate to the following file:
`\is\Samples\tbcmodel\TBC_Model_Unicode.xml`
5. Select the file and click **Import to Catalog**.
A message is displayed when the import is complete.

6. Click **OK**.

To import the Unicode sample metaoutline:

1. If necessary, select **File > XML Import/Export**.
The XML Import/Export dialog box is displayed.
2. Select the **Import** tab.
3. In the **XML File Path** text box, enter the file path for the Unicode sample metaoutline or click the **Open XML File** button and navigate to the following file:
`\\is\Samples\tbcmodel\TBC_Metaoutline_Unicode.xml`
4. Select the file and click **Import to Catalog**.
A message is displayed when the import is complete.
5. Click **OK**.
6. Click **Close** to close the **XML Import/Export** dialog box.

After setting up the sample application

You must connect to both TBC and TBC_MD from Integration Services Console to create, modify, and store TBC OLAP models and TBC metaoutlines. To make these connections, each database (TBC and TBC_MD) must be mapped to a supported ODBC driver, as described in Chapter 8, “Configuring relational data sources for DB2 OLAP Integration Server,” on page 63.

When you connect to Integration Services Console, you can view TBC columns, tables, OLAP models, and metaoutlines in Integration Services Console. For more information, see “Viewing TBC tables and columns.”

Note: You must connect to TBC and TBC_MD as user TBC unless you create user name aliases or synonyms in the RDBMS.

Viewing TBC tables and columns

After you set up the sample application and configure TBC and TBC_MD by mapping them to supported ODBC drivers, you can view TBC tables and columns in Integration Services Console

To see TBC tables and columns in the left frame of Integration Services Console:

1. Start the servers.
For instructions, see “Starting the Analytic Server on Windows” on page 25 or “Starting the Analytic Server on UNIX” on page 39.
2. Start Integration Services Console.
For instructions, see “Starting Integration Services Console” on page 57.
3. In a blank Integration Services Console window, select **Connections > OLAP Metadata Catalog > Connect**, and connect to the OLAP Metadata Catalog, TBC_MD.
4. In the **New** tab, double-click the **OLAP Model** icon.
5. In the **Data Source** dialog box, connect to the TBC sample database.

Viewing OLAP models

After you set up the sample application and configure TBC and TBC_MD by mapping them to supported ODBC drivers, you can view OLAP models in Integration Services Console.

To see the OLAP model (TBC Model) in the right frame of Integration Services Console:

1. Start the servers.
For instructions, see “Starting the Analytic Server on Windows” on page 25 or “Starting the Analytic Server on UNIX” on page 39.
2. Start Integration Services Console.
For instructions, see “Starting Integration Services Console” on page 57.
3. If the **Login** dialog box is not already displayed, in a blank Integration Services Console window, select **Connections > OLAP Metadata Catalog > Connect**.
4. Connect to the OLAP Metadata Catalog TBC_MD.
5. Click the **Existing** tab, select **TBC Model**, and click **Open**.
Hybrid Analysis: To view the Hybrid Analysis-enabled OLAP model, select **HA TBC Model**.
Unicode: To view the Unicode-enabled OLAP model, select **Sample_unicode**.
6. In the **Data Source** dialog box, connect to the TBC sample database.

Viewing metaoutlines

After you set up the sample application and configure TBC and TBC_MD by mapping them to supported ODBC drivers, you can view metaoutlines in Integration Services Console.

To see the metaoutline (TBC Metaoutline) in the right frame of Integration Services Console:

1. Start the servers.
For instructions, see “Starting the Analytic Server on Windows” on page 25 or “Starting the Analytic Server on UNIX” on page 39.
2. Start Integration Services Console.
For instructions, see “Starting Integration Services Console” on page 57.
3. If the **Login** dialog box is not already displayed, in a blank Integration Services Console window, select **Connections > OLAP Metadata Catalog > Connect**.
4. Connect to the OLAP Metadata Catalog TBC_MD.
5. Click the **Existing** tab, then expand the view of TBC Model and select TBC Metaoutline.
Hybrid Analysis: To view the metaoutline enabled for Hybrid Analysis, expand the view of **HA TBC Model** and select **HA TBC Metaoutline**.
Unicode: To view the metaoutline enabled for Unicode, expand the view of **Sample_unicode** and select **Sample_unicode_meta**.
6. Click **Open**.
The **Data Source** dialog box is displayed.
7. In the **Data Source** dialog box, connect to the TBC sample database.

Chapter 11. Upgrading and migrating from earlier versions

This chapter describes how to upgrade from an earlier release of DB2 OLAP Server or DB2 OLAP Integration Server to Version 8.2, and how to migrate your OLAP applications and databases from one computer to another.

It is recommended that you do a complete, not incremental, backup of all of your DB2 OLAP Server and DB2 OLAP Integration Server files before upgrading. You can then revert to the earlier version if you have problems.

This chapter contains the following topics:

- “Migrating DB2 OLAP Server”
- “Migrating DB2 OLAP Integration Server” on page 114
- “Other notes on migration” on page 115

Migrating DB2 OLAP Server

This section describes how to upgrade from an earlier version of DB2 OLAP Server to DB2 OLAP Server Version 8.2 on the same computer and describes what occurs during the upgrade process.

Supported API Versions

Programs compiled with the Version 8.2 Application Programming Interface do not work with earlier server releases. DB2 OLAP Server API programs compiled with header files from a specific release must use the libraries from the same release. For example, if the header files are from V7.1, you cannot use V8.1 run-time libraries. If your header files were from V8.1, you cannot use V8.2 runtime libraries.

Table 29 describes compatibility among API, runtime library, and Analytic Server versions.

Table 29. API Compatibility

API Header Files	Runtime Libraries	Analytic Server
DB2 OLAP Server V8.1 (Essbase 6.5.x)	DB2 OLAP Server V8.1 (Essbase 6.5.x)	<ul style="list-style-type: none">• DB2 OLAP Server V8.1 (Essbase 6.5.x)• DB2 OLAP Server V8.2 (Essbase 7.x)
DB2 OLAP Server V8.2 (Essbase 7.1.x)	DB2 OLAP Server V8.2 (Essbase 7.1.x)	<ul style="list-style-type: none">• DB2 OLAP Server V8.1 (Essbase 6.5.x)• DB2 OLAP Server V8.2 (Essbase 7.x)

Upgrading applications that use APIs

DB2 OLAP Server V8.2 (Essbase Release 7.1) introduces aggregate storage databases, which support multi-million member outlines. Compared to block storage databases, the number of members in aggregate storage databases may be several times larger. As a result, certain precautions are necessary when you

upgrade applications that use APIs that retrieve members from the server. For detailed information on the necessary precautions, see the section on migration in the *Analytic Services API Reference*.

Compatibility of non-Unicode applications between client and server software

V8.2 provides backward compatibility with earlier clients. This includes V8.1 (Essbase 6.5.x) and later of Application Manager, the Spreadsheet Add-in, and the MaxL Shell. It does not include Administration Services.

For information about release compatibility with Administration Services, see the *Administration Services Installation Guide*.

Understanding input/output defaults and upgrading

Read this section before upgrading to understand the two I/O access modes available, and how OLAP databases are affected by an upgrade in terms of cache sizes and the I/O access modes.

Determining which I/O access mode to use

Buffered I/O uses the buffer cache of the file system. Direct I/O bypasses the file system buffer cache and is able to provide faster response time and more potential to optimize cache sizes. Your databases might be using buffered I/O or direct I/O.

V8.2 (Essbase 7.x) and V8.1 (Essbase 6.5.x) databases use buffered I/O by default, unless `essbase.cfg` contains the setting `DIRECTIO TRUE`. When upgraded to V8.2, all V8.1 databases use the I/O access mode that was used in that release. The I/O access mode for any database can be changed, using the database settings.

The `DIRECTIO` setting is server-wide, and affects all databases. With Version 8, the access mode specified by `DIRECTIO` is only read once for each database, after upgrading or first creation of a database. Thereafter, the I/O access mode must be changed per database using the database settings.

Changing or preserving the I/O access mode

The I/O access mode is a database setting stored in the security file, rather than a server-wide `essbase.cfg` file setting that affects all databases. The `essbase.cfg` configuration-file setting `DIRECTIO TRUE|FALSE` is maintained for backward compatibility with V8.1. It is also used to provide a default value for newly created databases and for databases that are upgraded from an earlier release.

If a `DIRECTIO` setting exists in the `essbase.cfg` file at upgrade time, only newly created or upgraded databases will be affected by the setting. DB2 OLAP Server reads the I/O access mode specification from `essbase.cfg` once for each database, and copies that information to the security file (`essbase.sec`). Thereafter, if you want to change the I/O access mode used by any database, you must change it at the database level using the database setting.

The I/O access mode can be set from Administration Services Console, MaxL (`alter database set io_access_mode`), or programmatically using the Analytic Services Application Programming Interface.

If you set a database to use direct I/O, DB2 OLAP Server will attempt to use direct I/O the next time the database is started. If direct I/O is not available on the platform at the time the database is started, DB2 OLAP Server will use buffered

I/O, which is the default. However, DB2 OLAP Server stores the I/O access mode you selected as a setting in the security file, and will attempt to use that I/O access mode each time the database is started.

Understanding how cache sizes are affected by an upgrade

When you upgrade, your cache sizes for existing databases will not change. If you are currently using the default I/O access mode (direct), your cache sizes for existing databases are probably large, because direct I/O requires larger cache sizes. If, after you upgrade, you plan to use the default I/O access mode (buffered), you should reduce the cache size settings before upgrading or before you start the upgraded database.

The following list explains default cache sizes for upgrading DB2 OLAP Server databases, for each I/O access mode:

- The data cache default size is the same for direct I/O and buffered I/O: 3 MB.
- The index cache default size is smaller for buffered I/O than for direct I/O: 1 MB for buffered I/O and 10 MB for direct I/O.
- The data file cache default size is 32 MB when direct I/O is used. The data file cache is not used when buffered I/O is in effect.
- The size of index pages is fixed at 8 K, regardless of the I/O access mode. The index page size is unchangeable.

To override the defaults after upgrading, change the database settings or properties before you start the upgraded database for the first time.

For more information on buffered I/O and direct I/O, see the *Database Administrator's Guide*.

Upgrading databases to Version 8.2

This section provides migration details and tells you how to upgrade databases from earlier versions of DB2 OLAP Server to Version 8.2.

When does DB2 OLAP Server migrate the files?

DB2 OLAP Server migrates databases when the database is started. By default, a database is set to start when its application starts. The OLAP kernel checks for files resulting from previous unsuccessful migrations, restarting the migration if necessary.

DB2 OLAP Server migrates the `ESSxxxx.IND`, `dbname.ESM`, and the `dbname.TCT` files when the database is started. DB2 OLAP Server migrates the `ESSxxxx.PAG` file as data blocks are accessed; hence, the `.PAG` file migrates when you run the `VALIDATE` command after starting the database. After the kernel files are migrated, they are not backwards-compatible with the earlier release.

Steps for upgrading databases to Version 8.2

The following steps explain how to upgrade to Version 8.2 from an earlier release on the same computer. To move databases to another computer without upgrading, see "Moving applications and databases to another computer" on page 112. To upgrade and migrate databases to another computer, see "Upgrading and moving applications and databases at the same time" on page 113.

After migration, you can restore databases from earlier releases only from backups. Therefore, be sure to back up databases before starting to upgrade.

To upgrade to Version 8.2 on the same computer, proceed in the following order for each database:

1. Run the **alter database validate** command in MaxL against the database.
If the command returns errors, revert to a backup that is free of those errors.
2. Back up all application files, database files, and the security file.
3. If you are using LROs in a production environment, run the LISTLINKEDOBJECTS command in ESSCMD before upgrading. This command returns a list of LROs contained in the databases.
4. Stop the Analytic Server and Integration Server, if they are is running.
5. If you are installing over an existing Analytic Services installation, uninstall the existing release before installing V8.2.
Uninstalling will not remove the app folders, existing applications, the essbase.sec file, or the essbase.cfg file.
If you are upgrading on UNIX, you might need to uninstall manually. Back up the installation directory and then remove it before installing V8.2. After removing the installation directory, install Version 8.2, recreate applications and databases, and restore from the backup.
6. Install DB2 OLAP Server Version 8.2 to the same directory as the earlier DB2 OLAP Server installation.
7. Start the DB2 OLAP Server Agent (ESSBASE.EXE).
If you want to change database settings, this is a convenient point at which to do so. If you change the settings now, you will not have to restart the database to make the settings effective. See “Determining which I/O access mode to use” on page 110 for information on the default settings.
8. Select a database or load an application.
9. Run the VALIDATE command in ESSCMD against the migrated database. VALIDATE prompts you to specify a name for the error log file that it will use.
10. If the validation returns only LRO-related errors to the log file after upgrading, you must restore data from the earlier backup and re-create the LROs:
 - a. Either restore data from a backup of the database that does not contain LROs, or reload from a database export.
 - b. Restart the database in DB2 OLAP Server Version 8.2.
DB2 OLAP Server migrates the database to Version 8.2 format if the database was restored.
 - c. Run the **alter database validate** command in MaxL against the upgraded database.
 - d. Re-create the LROs, using the LISTLINKEDOBJECTS output as a guide. You may need to review the output from LISTLINKEDOBJECTS manually to verify its completeness.
11. Upon successful completion, unload the database and then back up the Version 8.2 database files.

Moving applications and databases to another computer

To move databases to a different computer, or to upgrade to Version 8.1 on a different computer manually, proceed in the following order for each database:

1. Run the **alter database validate** command in MaxL against the database you will migrate.

If the validation returns errors, revert to a backup that is free of those errors.

2. Back up all application files, database files, and the security file on the source server (that is, the server from which you are migrating).
3. Install DB2 OLAP Server on the target server computer.
4. Copy the ESSBASE.CFG file from the \BIN directory on the source OLAP server to the same directory on the target OLAP server using the file system.
5. On the target OLAP server, define disk volumes. To allocate a new volume, use **alter database** statement in MaxL and follow the prompts.

Consider carefully how you configure your disk volumes. Any changes you make to your disk volumes settings after you have loaded data on the new OLAP Server are reflected only in new data loads; changes are not retroactive.

6. Use the Migration Wizard in Administration Services to move the application to the target Analytic Server computer.

The names of the applications and database you create do not need to be the same as the ones on the source server. However, if you make changes to the names, make sure that these changes are reflected as necessary in script files, spreadsheet macros, and API-based applications. In addition, make sure that these changes are clearly communicated to the user base.

CAUTION:

Do not move the application directory to the target server through a file transfer via the operating system or via FTP.

7. Move any ESSCMD or MaxL scripts to the target server using the file system or via FTP. You can use the MaxL ESSCMD-to-MaxL script conversion utility.
8. Export data from the application on the source server in one of the following ways:
 - Choose All Data, Input Level Data, or 0 Level Data, depending on the size of your data set as well as how the database was initially loaded.
 - Use the initial data load files that were used to populate the database on the source server.
9. Import data to the new application on the target server.
10. Recalculate your database if:
 - You loaded input-level data from data load files, or
 - You exported only partial data from the database (for example, if you chose Level 0 or Input Level data only)
11. Repeat these steps for all other databases on the source server that you want to migrate to the new server.
12. Migrate security information by recreating user filters, groups, and permissions on the new server.

CAUTION:

Moving the security file (ESSBASE.SEC and its backup ESSBASE.BAK) between computers is not recommended or supported.

Upgrading and moving applications and databases at the same time

When you upgrade to a new release of DB2 OLAP Server, you may want to do so on a different computer. You have two options for upgrading and moving at the same time:

- First upgrade the existing installation to the new release on the source computer and then move applications and databases to the target computer. This alternative is the recommended order if you are upgrading and moving to a computer that runs a different operating system.

- First move existing applications and databases to the target computer, test the environment, and then upgrade to the new release on the target computer.

To move applications and databases to a different computer, use the Administration Services Migration Wizard. See the Administration Services Online Help for detailed information and instructions.

Migrating DB2 OLAP Integration Server

To help you create and execute your migration plan, this chapter provides information about platform support and hardware requirements, as well as planning for migrating OLAP outlines, databases, models, metaoutlines, and open database connectivity (ODBC) definitions.

Before you install DB2 OLAP Integration Server, back up all files from earlier releases. This will enable you to revert to an earlier release level if you encounter problems.

ODBC drivers on Windows

If you are using the DataDirect ODBC drivers supplied with a previous release of DB2 OLAP Integration Server on Windows, you must upgrade the drivers to version 4.2 when you install DB2 OLAP Integration Server Version 8.2. See “Supported ODBC drivers and relational data sources” on page 14 for a complete listing of the ODBC drivers supported per database and operating system.

Supported JDBC drivers for XML import/export

JDBC drivers are required to support the XML Import/Export functionality in DB2 OLAP Integration Server.

JDBC Type 1 is installed when you install the Java Runtime Environment. With the exception of Oracle, JDBC Type 1 is compatible with all RDBMSs supported by DB2 OLAP Integration Server. For information on supported JDBC drivers, see “Supported JDBC drivers for XML Import/Export in OIS” on page 16.

Migrating existing OLAP metadata catalogs

If you have an existing OLAP Metadata Catalog from a previous release of DB2 OLAP Integration Server, you must update it before using OLAP Metadata Catalog with the new version of the software.

CAUTION:

If you have installed the sample application from a previous release of DB2 OLAP Integration Server, you should back up the existing sample database, OLAP Metadata Catalog, and the OLAP models and metaoutlines stored in the catalog. You can then upgrade the existing catalog to be compatible with the current release of the software. You cannot, however, store new sample OLAP models and metaoutlines in the previous catalog.

To update an existing OLAP Metadata Catalog, perform the following tasks:

- Complete the upgrade steps for the existing OLAP Metadata Catalog described in “Upgrading the OLAP Metadata Catalog manually” on page 82.
- When connecting to the relational database, make sure that you use the same user name and password as you used when you created the original OLAP

Metadata Catalog. This procedure adds new tables to the OLAP Metadata Catalog or alters existing tables in the catalog without changing any existing data.

CAUTION:

After you update an OLAP Metadata Catalog, you cannot roll back to the previous release. The new release of OLAP Metadata Catalog is not compatible with previous releases of DB2 OLAP Integration Server. In addition, do not attempt to use the Version 8.2 catalog with previous releases of DB2 OLAP Integration Server. Doing so could result in corrupted OLAP Metadata Catalog data.

Other notes on migration

When currency databases are migrated, some of the database settings are not set, such as the number of dimensions, the currency time, and the currency category. Once the outlines and databases are migrated, these settings will be updated.

Users and groups can be migrated without applications and databases, but if the users or groups are added before the applications and databases, their association with those applications and databases must be manually updated.

Existing applications, databases, filters, users, and groups on the destination server are not overwritten or migrated. If they already exist, they will not be re-created.

The user must have supervisor privileges on the OLAP destination server.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Trademarks

1-2-3	Informix	AIX
Lotus	AIX 5L	OS/390
DB2	PowerPC	DB2 OLAP Server
Redbooks	DB2 Universal Database	RS/6000
IBM	Domino	Tivoli
z/OS	WebSphere	

The following terms are trademarks or registered trademarks of other companies:

Java is a trademark of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux[™] is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Index

Special characters

.cshrc, updating 37
.profile, updating 37
\$ODBCINI 46

A

abbreviations for product names viii
adding data source to odbcc.ini 68, 73, 77
Agent window
 using in AIX, Solaris Operating Environment, and HP-UX 43
 using in Windows 29
aggregate storage sample application 53
AIX server requirements, hardware and software 9
analytic server
 connecting 59
Analytic Server
 starting on UNIX 39
 starting on Windows 25
 stopping on UNIX 41
API
 compatibility with earlier releases 6
API client requirements, hardware and software 13
API, supported versions 109
application
 automating startup on Windows 26
Application Manager 6
ARBORPATH variable 3
ASOsamp application 51
authentication 30

B

Basic database 51
batch files
 for loading TBC database 92
BCP utility 89
books
 installing 24
 list of viii
building
 OLAP Metadata Catalog 79
 sample application 87

C

cache size, migration 111
checklist for installation 1
client and server compatibility 5
client requirements, hardware and software 12
client software
 RDBMS, required 64
client utility programs, database 88
columns, viewing in sample application 106

Command Center 83
command line utilities, SQL 88
Command Window, DB2 89
Company name, parameter
 for Windows 25
compatibility
 APIs 109
 earlier version 5
 servers and clients 110
configuring
 database client software 71
 ODBC (Open Database Connectivity)
 data sources 63
 ODBC computer location 63
 odbc.ini file 72
connecting 57
connection defaults 60
connection problems 42
creating
 database for TBC 91, 104
 database for TBC_MD 93
 ODBC server software data sources 63
 OLAP Metadata Catalog
 automatically 80
 OLAP Metadata Catalog
 manually 81
 OLAP Metadata Catalog tables 80
 OLAP Metadata Catalogs 79, 80
 sample application 87
 tables for TBC database 92
 TBC database 91

D

data loads
 with TBC sample application
 tables 92
data migration
 OLAP Metadata Catalog 82
data sources
 adding to odbcc.ini (UNIX) 68, 73, 77
 configuring 63
 configuring on UNIX systems 72
 configuring on Windows systems 64
 connecting 57, 60
 setting up TBC 91
 supported 14
database client software
 configuring 71
 information 64
database scripts
 for OLAP Metadata Catalog 83
 for TBC database 91
 for TBC_MD database 94
databases
 configuring for ODBC 63
 OLAP Metadata Catalog,
 requirements 80
 TBC_MD, creating 93
 TBC_MD, requirements 93
databases (*continued*)
 TBC, creating 91, 104
 TBC, requirements 91, 93, 104
databases, migrating 111
DB2
 -tvf utility 83, 89
 Command Center utility 83
 Command Window 89
 odbc.ini file example 74
DB2 OLAP Integration Server
 connecting 57
 installing 21, 35
 publications viii
DB2 OLAP Server
 automating startup on Windows 25
 connecting 57
 installing on AIX, Solaris Operating Environment, and HP-UX 33
 installing on Windows 19
 license 29, 43
 publications viii
 sample application 51
 stopping on Windows 26
DB2 UDB for z/OS and OS/390
 configuration parameter in eis.cfg (for UNIX) 77
 configuration parameter in eis.cfg (for Windows) 68
 default port number 66
DBAccess 83, 89
deleting
 OLAP Metadata Catalog 84
 OLAP Metadata Catalog tables 83
 TBC database tables 91
 TBC_MD database tables 94
Demo application 51
Desktop for OIS 57
dialog boxes
 Login 95
directories
 sample application scripts 90, 99
 TBC data load script 90, 97, 99, 100, 102, 105
 TBC table creation script 90, 97, 99, 100, 102
 TBC_MD data load script 90
 TBC_MD table creation script 90, 97, 99, 100, 102, 105
DMDemo application 51
documentation
 installing 24
 list of viii
Domino 31
dropping tables
 OLAP Metadata Catalog 83
 TBC database 91
 TBC_MD database 94

E

- editing
 - configuration for a data source 71
 - odbc.ini file 72
- eis.cfg
 - DB2 UDB for z/OS and OS/390
 - configuration parameter (for UNIX) 77
 - DB2 UDB for z/OS and OS/390
 - configuration parameter (for Windows) 68
- environment settings 36
- environment variables
 - for database clients 38
 - for ODBC on UNIX 72
- ESS_CSS_JVM_OPTION1 variable 42
- ESS_JVM_OPTION1 variable 42
- essbase.exe, running as a service 28
- essbaseenv.sh and essbaseenv.csh
 - environment scripts 37
- essjava.sh and essjava.csh environment scripts 37
- ESSLANG variable 4
- examples, odbc.ini files 74, 75, 76
- external authentication 30

F

- firewall
 - connecting through 59

H

- hardware requirements 7
- hisjava.sh and hisjava.csh environment scripts 37
- HP-UX server requirements, hardware and software 10
- Hybrid Analysis 45
 - enabling for Spreadsheet Add-in 48
 - enabling in manual sample application setup 87
 - library files on UNIX 46
 - required utilities for sample application setup 97
 - sample application for DB2 UDB 97
 - sample application for Oracle 99
 - sample application for SQL Server 100
 - sample application for Teradata 102
 - sample application setup 96
 - viewing the metaoutline 107
 - viewing the OLAP model 107
- HYPERION_HOME variable 3

I

- I/O access 110
- Informix
 - DBAccess utility 83, 89
 - example odbc.ini file 75, 76
- inst-sql.sh 46
- installing
 - OLAP Metadata Catalog 79

- installing DB2 OLAP Server
 - before starting on UNIX 33
 - before starting on Windows 19
 - getting started 7
 - installation steps for Windows 20
 - OLAP Miner 22
 - on AIX, Solaris, and HP-UX 34
 - on Windows 19
 - planning 1
 - requirements 7
- integration server
 - connecting 58
- Integration Server
 - starting on UNIX 41
 - starting on Windows 26
 - stopping on UNIX 42
- Integration Services Console
 - starting 57
- Integration Services Shell
 - starting 57
- Internl database 51
- interoperability 5
- ISQL utility 89

J

- Java
 - on Windows 27
- Java connection problems 42
- Java, environment variables 37
- JDBC 114
- JDBC drivers supported 16
- jrehome 27

L

- lc*.sql files 95
- ld*.sql files 92
- LDAP 31
- library files for relational integration 46
- library files on UNIX 46
- library paths on UNIX 48
- license 29, 43
- Linux server requirements, hardware and software 11
- loading data
 - into TBC database 92
 - into TBC_MD database using XML Import utility 95
- loading data into samples 55
- local client installation 19
- Login dialog box 95

M

- manuals
 - installing 24
 - list of viii
- Metadata Catalog
 - creating Unicode 81
- metadata catalogs
 - migrating 114
- migration
 - across servers 112
 - API compatibility 109
 - DB2 OLAP Integration Server 114

- migration (*continued*)
 - DB2 OLAP Server 109
 - I/O access 110
 - OLAP Metadata Catalog 82
 - planning for 2

N

- NT LAN Manager 31

O

- oc_create_*_unicode.sql files 94
- oc_create*.sql files 83
- oc_drop*.sql files 83
- oc_upgrade651*.sql files 83
- oc*_upgrd20.sql files 81, 82
- ocscrip directory, contents of 83
- ODBC
 - environment variables on UNIX 38
 - linking from SQL Interface 47
- ODBC (Open Database Connectivity)
 - configuring data sources 63
 - configuring, which computer 63
 - data sources, odbc.ini file heading 73
 - environment variables on UNIX 72
 - required database client software 64
- ODBC Administrator 64, 65
- ODBC drivers
 - required minimum level 14
- odbc.ini file
 - adding a data source to 68, 73, 77
 - bracketed headings 73
 - configuring data sources 72
 - DB2 example 74
 - editing 72
 - Informix example 75, 76
 - Oracle example 75
 - Teradata example 76
- OLAP Metadata Catalog
 - creating 79
 - creating automatically 80
 - creating tables for manually 81
 - data migration 82
 - database platforms supported 79
 - database requirements 80
 - deleting 84
 - deleting tables 81
 - information 79
 - post-setup steps 84
 - upgrading 82
- OLAP metaoutlines, viewing in sample application 106, 107
- OLAP Miner
 - installing on AIX 36
 - installing on Windows 22
- OLAP models, viewing in sample application 106, 107
- OLAPDoc.ini 24
- olapisvc, running as a service 28
- open database connectivity.
 - See ODBC (Open Database Connectivity)
- operations, managing server
 - commands for AIX, Solaris Operating Environment, and HP-UX 43

operations, managing server (*continued*)
 commands for Windows 29
Oracle
 odbc.ini file example 75
 SQL*Loader 92
 SQL*Plus 83, 89

P

partitioning sample application 53
PATH
 updating for Java on Windows 27
planning for installation 1
publications
 installing 24
 list of viii

Q

Query Analyzer utility (SQL Server) 83, 89

R

rebuilding tables 82, 83
reconnecting to DB2 OLAP Integration Server 62
relational data sources 14
relational databases
 client software required 64
relational integration 45
Remote Authentication Module 30
removing
 OLAP Metadata Catalog 84
 OLAP Metadata Catalog tables 83
 TBC database tables 91
 TBC_MD database tables 94
required hardware and software 7
runtime client 23

S

Sampeast application 51
sample application
 creating 87
 creating for Hybrid Analysis 96
 creating for Unicode 104
 for DB2 OLAP Server 51
 preparation steps 88
 required utility programs 88
 setting up automatically 89
 setting up manually 90
 viewing 106, 107
Sample database 51
sample database, TBC 87
Sample_U application 51
samples\tbcdbase directory, contents of 91
Samppart application 51
scripts
 for loading Hybrid Analysis data 92
 for loading TBC database 92
 for loading TBC_MD database 95
 for OLAP Metadata Catalog 83
 for TBC database 91

scripts (*continued*)
 for TBC_MD database 94
 location, sample application 90, 99
security 30
security file 4
server and client compatibility 5
server operations, managing
 commands for AIX, Solaris Operating Environment, and HP-UX 43
 commands for Windows 29
server requirements, hardware and software 7
servers 57
 connecting 57
 starting on UNIX 39
 starting on Windows 25, 26
service, running as 28
services on windows 20
shell scripts
 for loading TBC database 92
shell scripts for library paths on UNIX 48
silent install
 for documentation 24
 for most Windows components 23
software requirements 7
Solaris server requirements, hardware and software 8
sp-fkeys 84
SQL command line utilities 88
SQL Interface 45, 46
SQL scripts
 for loading TBC database 92
 for OLAP Metadata Catalog 83
 for TBC database 91
 for TBC_MD database 94
 location, sample application 90, 99
SQL Server
 Query Analyzer utility 83, 89
SQL utility programs, required for sample application 88
SQL*Loader 92
SQL*Plus 83, 89
startup for Windows
 applications, automatic startup 26
 DB2 OLAP server 25
 server, automatic startup 25
Sybase
 BCP utility 89
 execute permission required 84
 ISQL utility 83, 89
 sp-fkeys 84
 Sybssystemprocs 84
Sybssystemprocs 84
system administration
 commands for AIX, Solaris Operating Environment, and HP-UX 43
 commands for Windows 29
System password, parameter for Windows 25

T

tables
 creating for OLAP Metadata Catalog manually 81
 creating for TBC database 91

tables (*continued*)
 creating for TBC_MD database 94
 rebuilding tables 82, 83
 upgrading for OLAP Metadata Catalog 82
tables, viewing in sample application 106
TBC
 data load script location 90, 97, 99, 100, 102, 105
 database requirements 91, 93, 104
 database, creating tables for 91, 92
 database, viewing 106, 107
 loading data into 92
 sample database 87
 setting up relational data source 91
 table creation script location 90, 97, 99, 100, 102
 viewing columns in 106
 viewing metaoutline in 107
 viewing OLAP model in 107
 viewing tables in 106
TBC Metaoutline
 information 87
 viewing 106, 107
TBC Model
 information 87
 viewing 106, 107
TBC sample application 51
TBC_MD
 data load script location 90
 database requirements 93
 database, creating 93
 database, creating tables for 94
 loading data into using XML Import 95
 sample OLAP Metadata Catalog 87
 table creation script location 90, 97, 99, 100, 102, 105
TCP/IP 42
Teradata 47
 odbc.ini file example 76
Tivoli Directory Server 31
troubleshooting connection problems 42

U

unattended install on Windows 23
Unicode
 creating OLAP Metadata Catalog 81
 required utilities for sample application setup 104
 sample application for DB2 105
 sample application setup 104
 viewing the metaoutline 107
 viewing the OLAP model 107
UNIX
 environment variables, configuring 72
 ODBC data sources, configuring 72
 ODBC environment variables 72
upgrading
 OLAP Metadata Catalog 82
utility programs, required for sample application 88

V

- vi program 73
- viewing sample application 106, 107

W

- Windows 2000
 - ODBC Administrator 64
- Windows server requirements, hardware and software 7
- Windows services 20, 28
- Windows XP
 - ODBC Administrator 64

X

- Xcgrate database 51
- XML files for loading TBC_MD database 95
- XML Import for loading TBC_MD database 95

Y

- Your name, parameter for Windows 25



Program Number: 5724-OLP

Printed in USA

SC18-9205-00



Spine information:



IBM® DB2® OLAP Server™

DB2 OLAP Server Installation Guide

Version 8.2