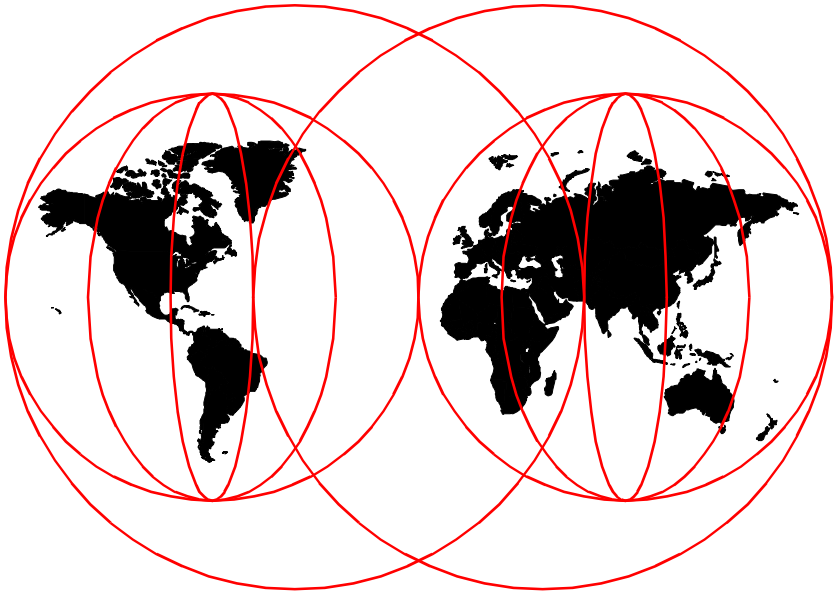
Getting Started with DB2 OLAP Server for OS/390

Seungrahn Hahn, Ashvin Amin, Klemens Dickhoever, Gerard Laumay



International Technical Support Organization

www.redbooks.ibm.com

SG24-5665-00

SG24-5665-00

International Technical Support Organization

Getting Started with DB2 OLAP Server for OS/390

March 2000

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix D, “Special notices” on page 231.

First Edition (March 2000)

This edition applies to DB2 OLAP Server for OS/390 Version 1, Release 1, Program Number 5640-OLP for use with the OS/390 Version 2.7 or later and DB2 UDB for OS/390 Version 6. DB2 OLAP Server for OS/390 Version 1.1 is based on Hyperion Essbase Version 5.0.2.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Tables

Preface

This redbook explores using the DB2 OLAP Server on the OS/390 system. It takes a close look at the tasks and considerations in installing, tailoring, and configuring the brand new DB2 OLAP Server for OS/390 Version 1.1. that is specific for the OS/390 environment, rather than the DB2 OLAP Server generic task. This redbook is designed to assist technical specialists and DB2 OLAP Server consultants in implementing this product for OS/390.

Chapter 1 contains an introduction to DB2 OLAP Server on OS/390, architectural overview, product components and their key features. This is especially useful for a person who needs to know about DB2 OLAP Server, is going to install it in the near future, or wants to understand the characteristics of OLAP tools.

Chapter 2 contains the information for planning the installation of the products. It describes the hardware and software requirements on the mainframe, the workstation, and the clients. It also describes the preparation for DB2, Web server, and security. Those who will install DB2 OLAP Server should read this chapter before installing the code.

Chapter 3 contains information on how to set up the OS/390 UNIX system environment for the DB2 OLAP Server. It covers HFS file management, customization of OS/390, TCP/IP, and security of OS/390 UNIX.

Chapter 4 contains the installation process for DB2 OLAP Server for OS/390. It has separate sections for multi-dimensional storage manager (MDSM) and relational storage manager (RSM). Based on your preference, you should focus on the relevant sections.

Chapter 5 contains the installation process for the administration and the end users. It covers Application Manager, Add-in features, Hyperion Wired for OLAP, and the Web Gateway.

Chapter 6 contains information for administration of DB2 OLAP Server. It covers activating and deactivating the OLAP Server, Essbase command, and backup and recovery of the data.

Chapter 7 contains OLAP database design on S/390. It covers design hints and tips of the cube design as well as physical design of the data to improve load and calculation processes.

Chapter 8 describes how to estimate the cube size, memory requirement, and CPU size using a sample cube model.

Chapter 9 contains overall considerations for security. This chapter should be read prior to installation.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization Poughkeepsie Center.

Seungrahn Hahn is a Business Intelligence specialist at the ITSO. Before joining the ITSO, she worked as an IT specialist in physical database design, performance tuning, and data sharing implementation with DB2 for OS/390, and BI application solutions in IBM Korea.

Ashvin Amin is a Senior technical specialist at IBM's Business Intelligence Solutions Center located in Dallas, Texas. His primary focus is to provide guidance in building and implementing BI solutions to customers in various industries. Ashvin has been with IBM for 16 years and has worked in the field as database administrator, performance tuning, logical and physical database design, and Decision Support/Business Intelligence solutions. He holds a Masters degree in Computer Science from City University of New York.

Klemens Dickhoever is a Senior Consultant and Project Manager at CGI Informatik Deutschland GmbH. CGI is a subsidiary of IBM Germany and is integrated into IBM Global Services. During the last year he has focused on BI. He has many years of experience in programming on mainframes, DB2, data modelling, application programming, and data warehousing. His technical background includes planning and realization of traditional transaction processing systems as well as designing, building, and managing data warehouse solutions.

Gerard Laumay is a S/390 Business Intelligence Architect at the Products and Solutions Support Center (PSSC) located in Montpellier, France. Part of the field support organization group, Gerard has been with IBM for 14 years, working in the field as an architect on S/390 environments for large French customers.

Thanks to the following people for their invaluable contributions to this project:

Alice Leung
IBM S/390 Business Intelligence Initiative Management Leader

Merrilee Osterhoudt
IBM S/390 Business Intelligence Technical Support Leader

Rich Conway
Robert Haimowitz
Vasilis Karras
International Technical Support Organization, Poughkeepsie Center

Terry Barthel
Alfred Schwab
Alison Chandler
Denny Sell
International Technical Support Organization, Poughkeepsie Center

David Share
IBM S/390 Briefing Center, S/390 Demonstrations

Karen Bradshaw
Jeff Gibson
David Goode
Gil Lee
Bill Malloy
John McPherson
Kay Parkin
An Phan
Gary Robinson
Leo Wu
IBM Santa Teresa Laboratories, DB2 OLAP Server Development

Iain Allen
Kevin P Jones
IBM BI SW Technical Sales, EMEA

Comments welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in “IBM Redbooks review” on page 251 to the fax number shown on the form.
- Use the online evaluation form found at http://www.redbooks.ibm.com/
- Send your comments in an internet note to redbook@us.ibm.com

Chapter 1. Introduction to the DB2 OLAP server

This chapter introduces the OnLine Analytical Processing (OLAP) technology concept. It describes the DB2 OLAP Server for OS/390 architecture and explains why OS/390 is a good platform on which to run the DB2 OLAP Server. Because the DB2 OLAP Server implements a client/server architecture, we describe the DB2 OLAP Server components available in the OS/390 environment and the client workstations needed to build a complete OLAP solution.

1.1 Introduction

In today's world, business strategies are built on the use and application of information and knowledge. As companies deploy enterprise-wide information-based analytical applications, real-time analysis of data and information are used by the entire enterprise, at every level, to maintain a competitive advantage.

IBM and Hyperion Solutions continue their successful partnership by working together to provide comprehensive sets of cross-industry OLAP solutions to help companies' leaders, teams, executives, knowledge workers, managers, and analysts make informed decisions. These solutions can be deployed on an enterprise scale or in a local department for a wide range of analytic applications, including product profitability and planning consolidations, sales and marketing analysis, forecasting, budgeting, and as an Executive Information System (EIS). These solutions take data from transactional systems, relational databases, legacy systems, spreadsheets, and data warehouses and transform it into business intelligence.

End users can choose from a wide variety of front-end tools with which to access information easily from all perspectives, seeing both the big picture and all levels of details. They can develop, share, and revise plans and forecasts with maximum speed; and they can access, manage, and schedule this information across the enterprise. With Hyperion Solutions and IBM products, corporations gain the information and agility they need to drive business performance.

1.2 What is OLAP

During the last ten years, a significant percentage of corporate data has migrated to relational databases. Relational databases have been used heavily in the areas of operations and control, with a particular emphasis on

transaction processing. Relational databases are also used for data warehouses. A *data warehouse* stores information that answers “who?” and “what?” questions about past events. In contrast to data warehousing, OnLine Analytical Processing (OLAP) uses a multidimensional view of aggregate data to provide quick access to strategic information for further analysis. OLAP enables analysts, managers, and executives to gain insight into data through fast, consistent, interactive access to a wide variety of possible views of information. OLAP transforms raw data so that it reflects the real dimensionality of the enterprise as understood by the user. OLAP enables decision-making to effect future actions.

OLAP and data warehousing are complementary. A data warehouse stores and manages data. OLAP subsets and transforms data warehouse data into strategic information. OLAP ranges from basic navigation and browsing (“slice and dice”), to calculations, to more serious analysis such as time series and complex modeling. As decision-makers exercise more advanced OLAP capabilities, they can use data access to gain knowledge rather than information only.

1.2.1 OLAP data attributes

At the risk of oversimplification, we will consider the properties of multidimensional data, data hierarchy, and usability.

The first and foremost property is the concept of *multidimensional data*. This concept is really an abstraction of relationships between data items. Entities of data in a multidimensional environment are usually referred to as “cubes”, which are composed of discrete cells of information. As an example, consider a database environment that tracks the sales of automobiles. Such a database could be constructed to have dimensions of automobile model, features, location of sale, date of sale, and measures (such as costs, prices, etc.). Each cell in this database stores a numeric value for a single combination of dimensional members, one from each of the five dimensions. With such a database, queries could be made on which colors sold best in which cities during which weeks.

Multidimensionality does not imply that all data entities must be stored in the database. They can be created dynamically from relational or non-relational data. This is the classic space-time trade-off; do we keep all of the relations in the database, or do we generate them on the fly? For implementations that statically contain complex relations, some form of sparsity management is necessary to keep the database size manageable.

The second property is the ability to view data hierarchically and view rollup calculations, derived data, and calculated data. This is often referred to as “drill-down” analysis. For example, in our automobile example, we might find that expensive red two-door automobiles sell well in Florida in the month of March. We would likely want to drill down, looking at the uniformity of the data. For example, we might discover specific weeks or specific cities that cause the trend. We might also discover that the trend is tied to specific features offered during a sales campaign.

Rollup calculations allow us to view more than the sum of detail data, such as the totals by state in our example.

The third property is *usability*. Many descriptions of OLAP go to great lengths to define specific attributes of usability such as client/server, query performance, multi-user support, and reporting characteristics. While some of these were perhaps advanced concepts during the gestation of OLAP, most are well understood and perhaps taken for granted today. Probably the most interesting attributes are the capabilities and limitations of the user interface and reporting capabilities. Some OLAP tools have attempted to extend SQL as a standard underpinning of OLAP tools while others have taken different approaches. Important reporting capabilities include point and click, drill down, drill up, pivot, and intuitive navigation.

OLAP databases are not intended to replace relational database technology as a repository for transactional data. As a consequence, the OLAP database usually needs to duplicate data contained in relational systems. Duplication allows reorganization and summarization of the data. This is consistent with data warehouse/data mart implementations in which the data is staged to ensure data consistency.

1.2.2 OLAP application areas

OLAP applications have been most commonly used in the financial and marketing areas, but as we show here, their uses do extend to other functions. Examples of OLAP applications deployed today include:

Marketing and analysis
Most companies require this kind of application. The consumer goods industry often has a large number of products and outlets and a high rate of change. Retailers have the potential to analyze huge amounts of data. Financial services industry customers (insurance, bank, etc.) need to analyze the product and customer profitability.

Database marketing
A specialized marketing application is taking advantage of analysis, combined with other statistical and data mining technologies (for example, who are the best customers for a targeted promotion?).

Budgeting
An OLAP tool can help in building budget applications.

Financial reporting and consolidation
OLAP helps to produce financial reports for management and legal requirements.

Management reporting
OLAP offers more flexibility by distributing interactive rather than static reports, providing a management reporting system that gives key performance indicators and critical success factors leading to better management of the organization.

Profitability analysis
OLAP helps in setting prices, deciding on promotional activities, selecting areas for investment or divestment, and anticipating competitive pressures.

Quality analysis
The need for consistent quality and reliability in goods and services is as great as ever.

Deploying OLAP applications across the enterprise benefits the whole organization and increases the productivity of business managers and developers. The inherent flexibility of OLAP applications means business users can become more self-sufficient, access strategic information more quickly, and take more-effective decisions using more-accurate data. IT developers also benefit from using OLAP software. They can deliver applications to business users faster, providing better service. Faster delivery of applications also reduces the application backlog.

Finally, OLAP provides the ability to model real business problems and a more efficient use of people resources. OLAP enables the organization as a whole to respond more quickly to market demands. Market responsiveness often improve revenue and profitability.

1.3 Various forms of OLAP

The multidimensional analysis tools (MDAT) market, also referred to as the OLAP market, consists of analysis tools that work from:

- Multidimensional file systems (MOLAP or OLAP)
- Relational star schemas (ROLAP)
- Hybrid stores (HOLAP), which are a combination of multidimensional and relational stores

By their nature, these tools are suited to different applications. MOLAP tools work best with highly summarized information to find broad trends and perform top-sheet analysis, and their strengths have traditionally been in financial reporting, financial consolidation, budgeting, and forecasting. As the technologies have become more scalable, these tools have also increased the breadth of their capabilities and are now also used, for example, in profitability analysis, revenue analysis, yield management, and supplier management.

ROLAP tools analyze existing relational warehouses in star schema or snowflake format. By working directly with the relational sources, ROLAP tools are well suited to provide detailed reports on large volumes of data.

Business intelligence (BI) is being distributed to an increasingly large set of users in enterprises. This is reflected in several technology trends. Vendors such as Cognos that have produced client-based business intelligence technologies are building server-based functions so that administration and distribution of BI objects are more efficient.

A corollary to the server implementation is the building of Web-based front ends for business intelligence.

The DB2 OLAP Server for OS/390 integrates two different storage managers: the multidimensional storage manager (MDSM), which uses the Hierarchical File System (HFS) to store the OLAP cube, and the relational storage manager (RSM), which uses DB2 as a relational database to store the OLAP data. DB2 OLAP Server for OS/390 is a multidimensional OLAP tool (MOLAP) that has the option to store the data in either a relational database or multidimensional files.

1.4 Why use OS/390 as a server platform for OLAP applications

A survey was taken with a set of customers, in June 1998 during the DB2 Technical Conference, to identify the benefits of the S/390 platform to run an OLAP product. The results are shown in Figure 1 on page 6. According to this survey, more than 72 percent of customers put a high value on getting an OLAP product on S/390.

According to customers, an S/390 OLAP product should offer advantages in:

- Analyzing their S/390 data directly on the S/390
- Using their S/390 skills to administer OLAP applications
- Using familiar desktop tools to analyze S/390 data
- Using the S/390 automation capability to run OLAP applications
- Using the graphical user interface tools to access multidimensional data on S/390

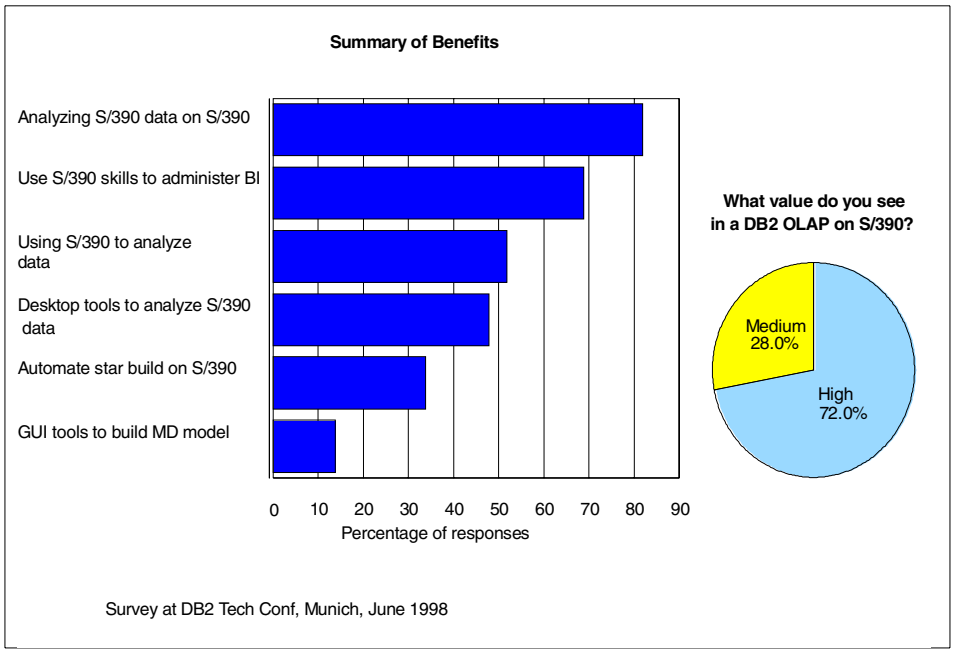


Figure 1. Main benefits of OLAP on S/390

A growing number of BI applications are becoming vital to the survival of an enterprise in an increasingly competitive world. You need to ensure that those applications that have become *mission-critical* are achieving the appropriate level of availability, performance, scalability, and manageability.

Running DB2 OLAP Server in the OS/390 environment provides the following advantages:

- Proven reliability and continuous availability.
- Ability to use the OS/390 Workload Manager (WLM) to manage workloads according to the business needs of the end users.
- Better concurrency between processes and the ability to run more calculations in parallel.

- Fast and efficient management of the data being processed, including backup/recovery and database reorganization.
- Proven, tested provision for recovery in the event of a disaster.
- High performance and the ability to handle multiple concurrent user queries.
- Ability to grow and change as the business requirements change.
- Low cost of ownership, including both acquisition and ongoing costs. You don't need to move your S/390 data to other servers for analysis. In reducing the number of servers and the complexity, you minimize your cost and effort in management.
- Effective leveraging of existing solutions, technology (hardware and software), processes, and skills to manage these new types of applications.
- OS/390 security benefit.

These are the main strengths that S/390 brings to the OLAP applications environment.

1.5 DB2 OLAP Server for OS/390 overview

The IBM DB2 OLAP Server for OS/390 is an online analytical processing (OLAP) server that you can use to create a wide range of multidimensional planning, analysis, and reporting applications.

It is the result of a joint venture between IBM and Hyperion Solutions Corporation, whose Essbase OLAP engine is used by the DB2 OLAP Server for OS/390. The product includes all capabilities of the Essbase product. In addition to HFS file systems, it offers the option of storing multidimensional databases as sets of relational tables using DB2 for OS/390.

The DB2 OLAP Server for OS/390 supports multi-user read/write access, large-scale data capacity, robust analytical calculations, and sophisticated OLAP queries. It provides intuitive multidimensional data navigation, which reduces the time and effort involved in training new users, as well as consistent, rapid response times in network-centric computing environments.

End users can use a wide range of client interface products that support Essbase API. They can analyze data from different points of view, using the drill-down, and slice and dice techniques that make the OLAP technology so powerful. The success of OLAP applications depends on the ease of use to set up new applications with a very high return on investment.

Figure 2 gives a sample list of end-user tools supported by the DB2 OLAP Server for OS/390.

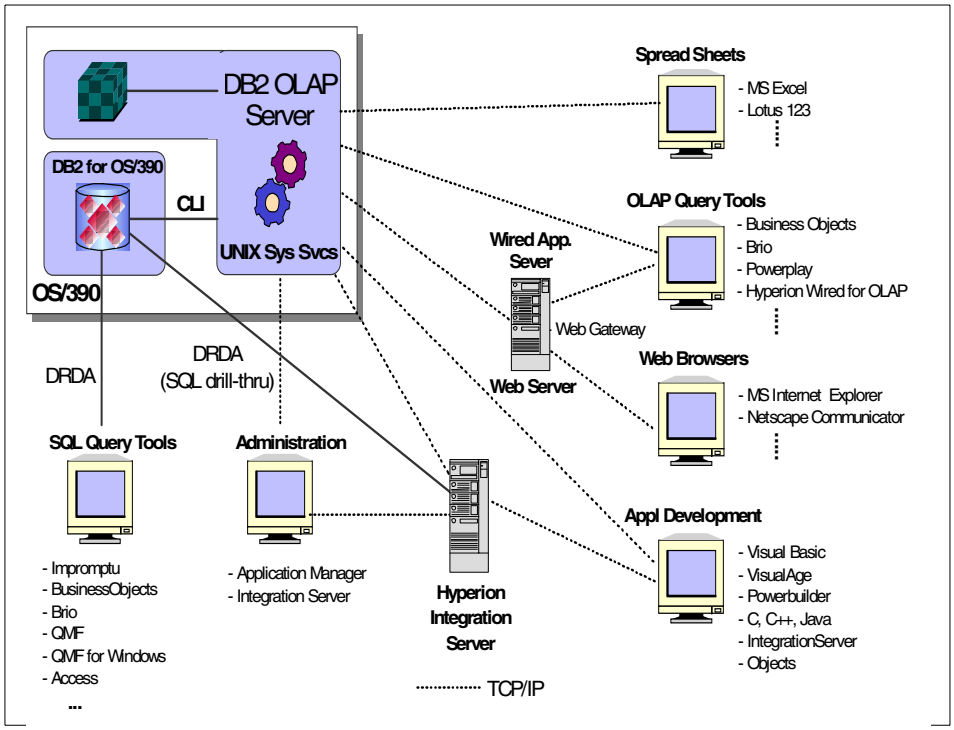


Figure 2. DB2 OLAP Server for OS/390 product overview

- DB2 OLAP Server for OS/390 operates in a client/server environment as a server to Essbase clients. When the relational storage manager (RSM) interface is used, it is also a client to DB2 for OS/390.
- On OS/390, the DB2 OLAP Server runs in an address space. Each started DB2 OLAP application creates a new address space. Users connected to the same opened application run as threads in the same OS/390 address space.
- Only TCP/IP is supported as a communication protocol between the DB2 OLAP Server for OS/390 engine and client machines.
- Distributed Relational Database Architecture (DRDA) connectivity is needed to access DB2 databases on OS/390 when using the SQL Drill-Through function without accessing the DB2 OLAP Server or using SQL query tools.

- The DB2 OLAP Server, initially available on NT, AIX, Sun Solaris, and HP-UX platforms, has been ported onto the OS/390 UNIX platform to take advantage of the S/390 strenghts.
- The administration of DB2 OLAP Server is done using the Application Manager interface.
- The DB2 OLAP Server fully supports the Essbase API and thereby all Essbase API-ready tools. DB2 OLAP Server for OS/390 integrates with more than 50 tools and application modules from Essbase partners.
- The DB2 OLAP Server supports existing Hyperion query tools such as Hyperion Wired for OLAP and Reporter.

1.6 DB2 OLAP Server components

This section provides an overview of the main server and workstation components of DB2 OLAP Server for OS/390.

Figure 3 on page 10 shows the general DB2 OLAP Server for OS/390 architecture, including the components to build an OLAP solution.

1.6.1 DB2 OLAP Server for OS/390 server components

DB2 OLAP Server for OS/390 Version 1.1 components include:

Essbase OLAP engine
DB2 OLAP Sever for OS/390 uses the Essbase OLAP engine for application design and management, data access and navigation, data load, data calculation, and application programming interfaces (APIs).

The OLAP engine can be used with all Essbase-ready front-end tools and applications developed by Hyperion and Hyperion partners.

Sample applications are provided with the server code.

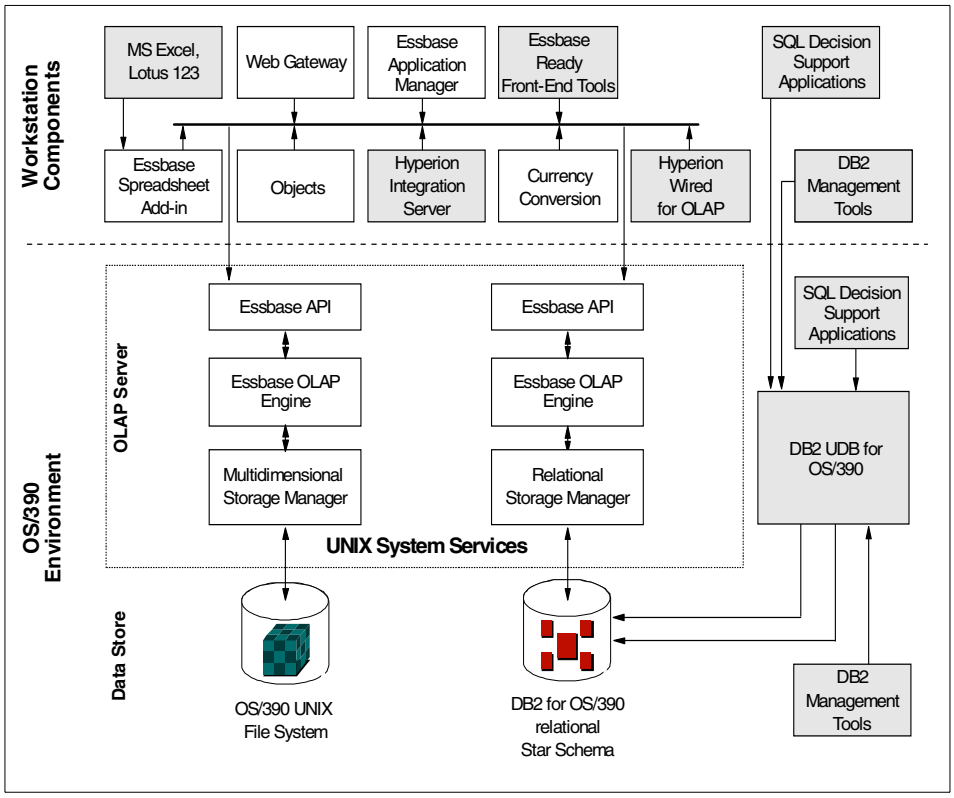


Figure 3. DB2 OLAP Server for OS/390 architecture

Relational storage manager (RSM)

The DB2 OLAP Server offers a relational storage manager (RSM) interface for the Essbase applications using DB2 for OS/390 as familiar relational database management system (RDBMS) management, backup, and recovery tools.

Using the RSM interface, DB2 OLAP Sever for OS/390 stores data in a relational database using a star-schema data structure. The end user can access the data using an Essbase client such as a spreadsheet or specific front-end tools such as Business Object, Brio, Cognos, and many others, or using standard SQL statements.

The RSM automatically creates and manages the necessary relational tables, views, and indexes within the star-schema.

Multidimensional storage manager (MDSM)
DB2 OLAP Server for OS/390 offers a multidimensional storage manager (MDSM) interface for applications where performance is the critical requirement. Using MDSM, data is stored into an OS/390 UNIX file system.

MDSM and RSM interfaces cannot run at the same time in the same OS/390 system image, but you can run multiple OLAP servers in different OS/390 system images running in a sysplex configuration.

DB2 OLAP Server for OS/390 Partitioning Option
The Partitioning Option for DB2 OLAP Server for OS/390 allows the administrator to divide OLAP applications and distribute them into separate physical partitions. Each partition can independently be an MDSM or RSM installation. You can load and calculate partitions in parallel on symmetric multiprocessing (SMP) computers, and distribute partitions across the network. MDSM and RSM are interoperable.

Partitioning is a powerful capability that allows you to create much larger OLAP applications, and to extend analyses outside of individual applications and across the enterprise.

1.6.2 DB2 OLAP Server workstation components

The following features are shipped with the DB2 OLAP Server for OS/390 product:

Essbase Application Manager
The Application Manager is a client/server architecture that can maintain servers, applications, and databases from a single administration point.

The Application Manager includes tools for building and modifying OLAP structures, defining database partitions loading data, creating analytical calculations, and controlling server processing. Graphical tools provide easy management of centralized or distributed databases, metadata synchronization, and data replication. The Application Manager also includes tools for administering users and user groups, managing security, and monitoring performance.

The Application Manager functions include:

- Metadata editor
- Calculation
- Data loading
- Partition editor
- Performance tuning

- User management
- Security management

You need to install the Essbase Application Manager component on a Windows workstation to administer DB2 OLAP Server for OS/390.

Essbase Spreadsheet Add-in
The Spreadsheet Add-in turns familiar desktop spreadsheets, Microsoft Excel, and Lotus 1-2-3, into tightly integrated DB2 OLAP Server clients. Users can leverage the power of desktop spreadsheets with the capacity, performance, and rich analytical capability of DB2 OLAP Server for OS/390.

1.6.3 Components that can be ordered with DB2 OLAP Server 1.1

In this section, we describe five products in the Tools Bundle feature that is a licensed product, and two optional products that can be run with DB2 OLAP Server at a workstation platform.

SQL Interface
DB2 OLAP Server SQL Interface integrates DB2 OLAP Server applications with relational data sources. It facilitates loading of both data and metadata directly into DB2 OLAP Server for OS/390. SQL Interface includes drivers that provide access to DB2 and work with additional data sources through IBM Data Joiner.

Currency Conversion
DB2 OLAP Server Currency Conversion allows you to model, analyze, and report on the effects of currency fluctuations and changing exchange rates on your business. Applications built with DB2 OLAP Server Currency Conversion automatically convert currency for budgeting, forecasting, reporting, financial consolidation, and other analytical applications.

Application programming interface (API)
DB2 OLAP Server provides an API consisting of a comprehensive library of more than 300 functions. This API is a reliable platform for building enterprise OLAP applications and is used by nearly 300 Essbase-ready applications, services, and tools partners, and by Hyperion for its own products.

Extended Spreadsheet Toolkit
Essbase Extended Spreadsheet Toolkit is a comprehensive library of more than 30 spreadsheet macros and 50 Visual Basic for Applications (VBA) functions that allow you to incorporate customized DB2 OLAP Server data navigation and analysis features into Microsoft Excel or Lotus 1-2-3 applications. Essbase Extended Spreadsheet Toolkit lets you extend

spreadsheets into custom OLAP application development and deployment environments.

SQL Drill-Through
Despite the inherent benefits of the multidimensional database for storing analytical data, some data elements that are required for analysis are better suited to remain in the relational structure. The SQL Drill-Through, which requires the SQL Interface, lets you access data in a relational data source from Essbase Spreadsheet Add-in, by mapping data from a DB2 OLAP Server multidimensional database to a relational database. SQL Drill-Through accesses the relational data source through an Open Database Connectivity (ODBC) driver. It lets you navigate from analytic data in DB2 OLAP Server databases to detail data in relational databases. It provides intuitive data navigation into a relational database by defining mappings between the dimensional attributes of a DB2 OLAP Server database and the fields of relational database tables.

The Web Gateway
The Web Gateway is a development platform for building Web-based HTML front ends for OLAP analytic applications. It enables high-speed, interactive read-write access to DB2 OLAP Server for OS/390 across the World Wide Web. Web Gateway delivers a comprehensive Web-based solution for ad-hoc multidimensional analysis as well as sophisticated management reporting.

The Web Gateway is an optional feature.

The Objects
Objects are a family of OLAP-aware components for developing robust online analytical processing (OLAP) applications.

Objects provide a set of open, ActiveX, OLAP-aware objects, letting users easily and rapidly develop customized OLAP applications on Windows 95/98/NT clients with minimal programming. Objects works with Microsoft Visual Basic and Microsoft C++.

Objects applications can be deployed in a wide range of computing environments: stand-alone, client-server, or distributed processing.

The Objects is an optional component delivered with the DB2 OLAP Server Objects product.

1.6.4 Additional components resold by IBM

In this section, we describe two product components of Hyperion that can be resold by IBM.

Hyperion Integration Server

Hyperion Integration Server (HIS) is a suite of graphical tools and scalable data integration services that dramatically reduce the time and expense to create, deploy, and manage analytical applications. Hyperion Integration Server uses centralized, reusable metadata to automate the process of creating and managing analytical applications from relational data sources such as data warehouses, data marts, transaction processing applications, and enterprise resource planning (ERP) systems using DB2 OLAP Server. Hyperion Integration Server allows users to seamlessly navigate from summarized, calculated, and derived data managed by DB2 OLAP Server to detail data stored in relational sources.

Hyperion Integration Server includes a shared enterprise OLAP catalog. End users and IT professionals can combine and customize objects stored in the OLAP catalog and then invoke Hyperion Integration Server to quickly and automatically create new analytical applications that are tailored to satisfy specific business requirements from relational data sources.

Hyperion Wired for OLAP

Hyperion Wired for OLAP is an OLAP analysis, presentation, and reporting solution for the enterprise. It is extremely easy to use, with intuitive navigation and highly interactive graphical displays that enable the widest range of users to quickly analyze critical business information. It gives users point-and-click access to advanced DB2 OLAP Server features. It uses a three-tier architecture with Windows and Java clients that allows organizations to deliver information to large user communities.

1.7 DB2 OLAP Server for OS/390 product packaging

DB2 OLAP Server for OS/390 Version 1.1 is shipped with the following components:

- DB2 OLAP Server running on OS/390 including:
  - The OLAP engine with MDSM and RSM interfaces
  - The Essbase API
- Client support running on 32-bit Windows connecting to the host via TCP/IP. The client support includes:
  - The Essbase Spreadsheet Add-in for Microsoft Excel
  - The Essbase Spreadsheet Add-in for Lotus 1-2-3
  - The Essbase Application Manager
- The Partitioning Option feature

DB2 OLAP Server for OS/390 includes one concurrent OLAP user license. Additional concurrent user licenses can be ordered using the DB2 OLAP Server Standard Edition Version 1.1 Use Entitlements.

In addition, features of the DB2 OLAP Server Standard Edition Version 1.1 for workstations can be ordered for use with DB2 OLAP Server for OS/390. A complete configuration involves components from DB2 OLAP Server for OS/390 and from the workstation product. This includes:

- DB2 OLAP Server Web Gateway 1.1
- DB2 OLAP Server Objects 1.1
- DB2 OLAP Server Tools Bundle 1.1, which includes:
  - SQL Interface
  - SQL Drill-Through
  - Currency conversion
  - Application Programming Interface (API)
  - Extended Spreadsheet Toolkit

Additional Hyperion products resold by IBM can complete the configuration:

- Hyperion Wired for OLAP
- Hyperion Integration Server

The DB2 OLAP Server installation license program prompts you to confirm the features you have purchased.

1.8 Multidimensional versus relational models

This section is intended to help you choose the storage interface to use with DB2 OLAP Server for OS/390 that will best fit your needs.

The product gives customers a choice to implement either a multidimensional storage manager (MDSM) or relational storage manager (RSM) method at install time.

In the MDSM implementation, the OLAP cube (applications) is stored in Hyperion's highly optimized proprietary file structure. The data and index components are stored in HFS files, but can only be accessed by Essbase-ready tools. Hyperion has published their API, so other software vendors can build tools to access the data stored in an Essbase OLAP cube. Currently there are more than 50 tools available to access data stored in an Essbase OLAP cube. MDSM creates two files to store data and index components.

The RSM method offers some other benefits. In this implementation, data and indexes are stored in DB2 relational tables. The data is stored in *fact* and *dimension tables* using the star schema format. Since the data is stored in relational tables, users can use structured query language (SQL) tools to access OLAP cubes, including Query Management Facility (QMF). Since the data is stored in DB2 relational tables, system administrators can take advantage of DB2 utilities for backup and recovery. All necessary DB2 tables/views are created by DB2 OLAP Server as the OLAP model is built. System administrator/database designers do not have to write any Data Definition Language (DDL) statements to create and maintain DB2 objects.

The following criteria can influence your choice between RSM and MDSM:

- If performance for load and calculation processes is important, then you should consider MDSM.
- If you have enough time during your batch window, choose RSM. If your batch window is constrained, then choose MDSM.
- If your primary interest is simplicity, then consider MDSM.
- If your primary interest is using your RDBMS (DB2) for OLAP applications to leverage the DB2 skills and administrative tools (backup/recovery, DB2 utilities), then choose RSM.
- If you are interested in data aggregation before loading, then choose RSM.
- If your batch update frequency is high (daily or weekly), choose MDSM. For monthly updates, choose RSM.

Summary

- Average query response time should not to be considered as a key decision factor between MDSM and RSM.
- Choose the multidimensional store MDSM when performance is a critical requirement for your application.
- Choose the relational store RSM for flexibility, such as the relational access from query tools, and the ability to leverage your administrative processes for database management.

After you have selected the right storage interface you want to install, go to the appropriate installation section. Section 4.2, “Post-SMP/E installation process for MDSM” on page 54 discusses MDSM installation and 4.3, “Post-SMP/E installation process for RSM” on page 69 explains RSM installation.

1.9 ITSO environment

We installed the DB2 OLAP Server at the ITSO Poughkeepsie. All examples shown in this redbook are based on the system configuration we used.

Our environment included the OS/390 server and Windows NT workstations. We had access to OS/390 Version 2 Release 7 including OS/390 Security Server (RACF), and DB2 UDB for OS/390 Version 6.1 running in a 9672 G6 logical partition (LPAR).

Hardware configuration for the S/390 server:

- 9672-X77 LPAR, including an OSA-2 adapter for LAN connectivity
- 512 MB central storage
- 3390 DASD

Software configuration for the S/390 server:

- OS/390 Version 2 Release 7
- OS/390 Communication Server (TCP/IP)
- OS/390 Security Server (RACF)
- WebSphere Application Server for OS/390
- DB2 UDB for OS/390 Version 6 Release 1
- DB2 OLAP Server for OS/390 Version 1.1

We installed the DB2 OLAP Server client environment on three NT workstations, connected to the OS/390 server.

Hardware configuration for the NT workstations:

- Netfinity 3000
- 196 MB RAM
- 2.3 GB disk drive

Software configuration for the NT workstations:

- Windows NT 4.0
- Lotus 1-2-3 R9.5 and Microsoft Excel 2000 spreadsheets
- Netscape Communicator 4.6
- Essbase Application Manager 5.0.2
- Hyperion Wired for OLAP Administrator 4.1.0
- Hyperion Wired for OLAP Analyzer 4.1.0 Windows
- Hyperion Wired for OLAP Analyzer 4.1.0 Web
- Hyperion Wired for OLAP Application Server 4.1.0
- Hyperion Integration Server Version 1
- DB2 OLAP Server Web Gateway

- IBM Apache Web Server 1.3

Figure 4 shows an overview of the DB2 OLAP Server architecture we used.

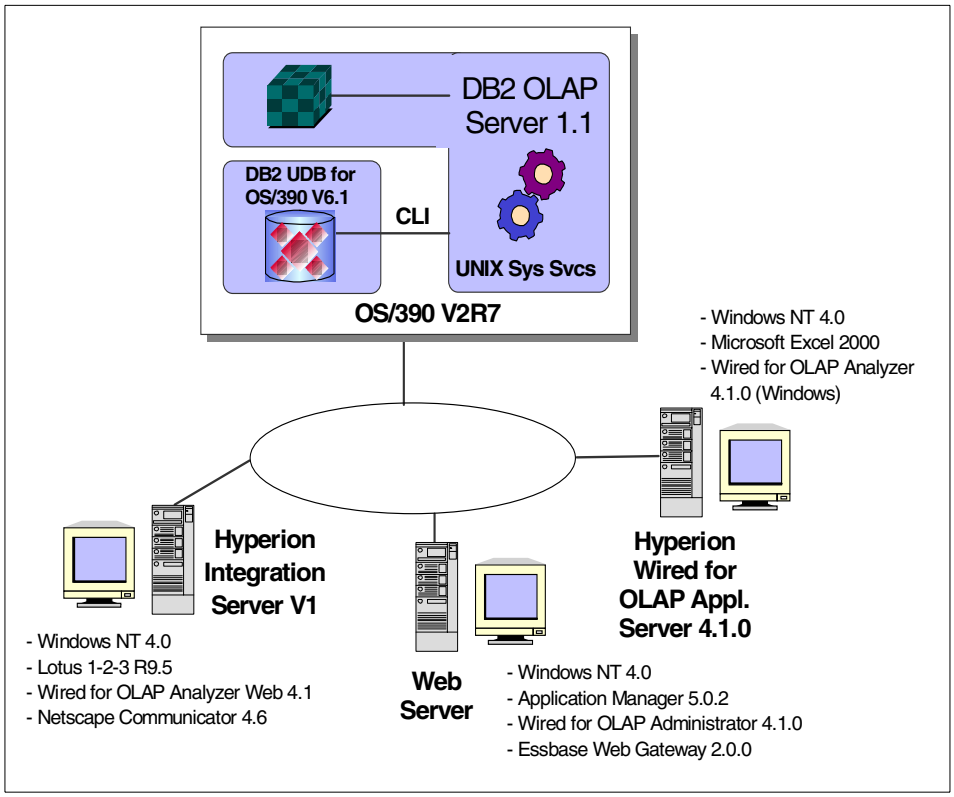


Figure 4. DB2 OLAP Server architecture at the ITSO

We defined a workstation for DB2 OLAP Server administration and two workstations as end-user platforms, one for the Window user and the other for the Web browser user.

Chapter 2. Planning for installation

This chapter presents the hardware and software requirements to install DB2 OLAP Server for OS/390 in an OS/390 environment, the tools on other servers outside S/390, when you need to build a 3-tier architecture, and client workstation tools available from IBM and Hyperion Solutions Corporation.

2.1 Hardware and software requirements

This section describes the hardware and software requirements needed to perform the installation of DB2 OLAP Server for OS/390 in your environment.

We describe the environment needed on the server side and the client workstations.

2.1.1 S/390 Server hardware requirements

Processor requirement

DB2 OLAP Server for OS/390 can be installed on any hardware environment that supports OS/390 Version 2 Release 5 or higher.

To estimate the size of the CPU needed to run DB2 OLAP Server, refer to Chapter 8, “Capacity planning” on page 193.

Disk space requirement

DB2 OLAP Server libraries can reside on any DASD supported by the required software. The total space values provided in the table below are for 3390 DASD.

Table 1. Total DASD space required by DB2 OLAP Server for OS/390

Library type	Total space required	Kilobytes
Target	25 blocks (each block is 512 bytes)	12.5
Distribution	970 blocks	485.0
HFS	104,506 blocks	52,253
Total including TAR file	292,166 blocks	146,083

You may have to consider adding extra space depending upon your application data organization requirements.

If you use the MDSM interface, you will have to store your application data and all Essbase application files (Essbase outline files, data load rules files, calculation scripts, ASCII text files, etc.) in the OS/390 UNIX file system. We strongly recommend that you allocate your application data in a separate file system mounted as a subdirectory of the *app* directory that contains your data. Depending on the size of your application data, you can create one HFS data set per application, or only one HFS file system for all your applications stored in the *app* directory.

Do not forget to mount this directory to your Essbase directory and to modify the SYS1.PARMLIB(BPXPRMxx) member to automatically mount this MVS HFS data set upon each IPL.

If you use the RSM interface, your data will be stored into a DB2 database created for the DB2 OLAP Server. Nevertheless, all Essbase application definition files (Essbase outline files, data load rules files, calculation scripts, ASCII text files, etc.) reside in the OS/390 UNIX file system *app* directory. This means you should estimate the total space needed to handle all these files according to your application environment and create a separate HFS data set if needed.

2.1.2 S/390 Server software requirements

The following software is required to run DB2 OLAP Server for OS/390:

- OS/390 Version 2.5 (5647-A01) or higher. OS/390 Version 2.7 or higher is strongly recommended.
- OS/390 Security Server (RACF) or an equivalent product.
- DB2 for OS/390 Version 5.1 (5655-DB2) or DB2 UDB for OS/390 Version 6.1 (5645-DB2) to run DB2 OLAP Server using the relational storage manager (RSM) interface.

In all cases, refer to the PSP bucket to check the last maintenance level delivered by IBM to perform this installation.

If you are running OS/390 Version 2.5 or Version 2.6, in order to avoid syntax errors during the SMP/E installation on the DESCRIPTION operand introduced with OS/390 Version 2.7, you must first install PTF UR51068. With OS/390 V2R7, DB2 OLAP Server can benefit from the new features added for the OS/390 UNIX and performance improvements. OS/390 Version 2.7 or higher is recommended for the installation of DB2 OLAP Server.

DB2 OLAP Server communicates with other components on OS/390 and Essbase client machines using TCP/IP. The TCP/IP feature integrated in the

OS/390 Communication Server or an equivalent product is mandatory. The DB2 OLAP Server engine always listens for incoming requests on port 1423.

If you want to run the DB2 OLAP Server relational storage interface (RSM) or SQL Interface (that is part of the Tools Bundle in DB2 OLAP Server V1.1), DB2 for OS/390 Version 5 Release 1 (5655-DB2) or higher is required. Service level 9802 is mandatory to perform the installation. DB2 OLAP Server uses the Call Level Interface (CLI) to communicate with DB2. The following PTFs and subsequent actions are required:

- Up to UQ18719
- UQ30999 for DB2 Version 5 or UQ31000 for DB2 UDB Version 6.
- UQ37985 for DB2 for OS/390 Version 5 or UQ37982 for DB2 UDB for OS/390 Version 6 for SQL Drill-Through support
- UQ31080 for DB2 V5 or UQ31081 for DB2 V6 for code conversion. After applying this PTF, you must run DSNTEJ1T job in hlq.NEW.SDSNSAMP that is created by the PTF.

Updated information on support level and PTFs for DB2 is available from the Web at:

```
http://www.software.ibm.com/data/db2/db2tech/
```

2.1.3 Workstation hardware requirements

Different options are available for the client side. Table 2 shows the recommended hardware requirements for installing the various components.

Table 2. Workstation hardware requirements

Product or component	Hardware requirements
Spreadsheet Add-in for Lotus 1-2-3 or Microsoft Excel	Pentium processor 32 MB RAM (minimum) 18 MB disk space Peripheral: CD-ROM drive
Application Manager	Pentium processor 32 MB RAM (minimum) 20 MB disk space Peripheral: CD-ROM drive
SQL Drill-Through	1.35 MB disk space Peripheral: CD-ROM drive

Product or component	Hardware requirements
API Client for Windows environment	Pentium processor for Windows 95/98/NT 16 MB RAM for Windows 95/98, or 32 MB RAM for Windows NT 32 MB disk space Peripheral: CD-ROM drive
API Client for UNIX environment	RS/6000 for AIX, or Sun SPARC/ULTRASPARC for Solaris, or HP PA-RISC for HP-UX 64 MB RAM 12 MB disk space Peripheral: CD-ROM drive
Web Gateway	Pentium processor for Windows, or Sun SPARC/ULTRASPARC for Solaris Minimum of 15 MB disk space Peripheral: CD-ROM drive
Objects	Pentium processor 32 MB RAM 20 MB disk space Peripheral: CD-ROM drive
Hyperion Wired for OLAP for Server platform	Pentium processor, or RS/6000 PowerPC 64 MB RAM 300 MB disk space
Hyperion Wired for OLAP for Client platform	Pentium processor 16 MB RAM 10 MB disk space
Hyperion Integration Server	**Server platform:** Pentium processor, or RS/6000 PowerPC 64 MB RAM recommended 300 MB disk space **Client platform:** Pentium processor 32 MB RAM 89 MB disk space

2.1.4 Workstation software requirements

Table 3 lists the software requirements for installing the various client components.

Table 3. Workstation software requirements

Product or component	Software requirements
Spreadsheet Add-in for Lotus 1-2-3	Windows 95/98 or Windows/NT 3.51 or 4.0 Lotus 1-2-3 5.0, 95/97, or Release 9 TCP/IP
Spreadsheet Add-in for Microsoft Excel	Windows 95/98, or Windows NT 3.51 or 4.0 Microsoft Excel 5.0, Excel 95, Excel 97, or Excel 2000 TCP/IP
Application Manager	Windows NT 4.0 or Windows 95/98
SQL Drill-Through	Note: We tested SQL Drill-Through only with the following environments: Windows NT 4.0, Excel 2000, Lotus 1-2-3 Database source: DB2 UDB for OS/390
API Client for Windows environment	Windows 95/98, or Windows NT 3.51 or NT 4.0
API Client for UNIX environment	AIX 4.2 or later; or Sun Solaris 2.5.1 or 2.6; or HP-UX 10.20 or 11.0
Extended Spreadsheet Toolkit	Windows 95/98, or Windows NT
Web Gateway: Web Server with Web Gateway for Windows	Windows NT Server 3.51 (Service Pack 3) or later, Microsoft Internet Information Server 3.0 for Windows NT or higher, Microsoft Merchant Server **Communications:** TCP/IP **Web Browsers:** Microsoft Internet Explorer 3.0 or higher Netscape 3.0 or higher

Product or component	Software requirements
Web Gateway: Web Server with Web Gateway for Solaris	Solaris 2.5.1 or 2.6, Netscape Communications Server, Netscape Commerce Server 2.0 or higher **Communications:** TCP/IP **Web Browsers:** Microsoft Internet Explorer 3.0 or higher Netscape 3.0 or higher
Objects	**Operating System:** Windows 95/98/NT 4.0 **Communication:** TCP/IP **Programming Environment:** Microsoft Visual Basic 4 or later Microsoft Visual C++ 4.2 **Web Deployment** ActiveX-compatible Web browsers **DB2 OLAP Server 1.1**
Hyperion Wired for OLAP	**Server platform for Wired Application Server:** Windows NT 4.0 or later (NT Server recommended), or AIX 4.2 Microsoft IIS, or Netscape Enterprise as Web Server for Web browser support or any other Web server **Server platform for Repository:** Windows 95/98/NT SQL Server 6.7 or 7.0, DB2 UDB, or Oracle 7.3 or 8i for Wired Enterprise Repository **Client platform for Wired Analyzer:** Windows 95/98/NT 4.0 or later Microsoft Explorer 4.0, or Netscape Communicator 4.05 or later for Web Viewer
Hyperion Integration Server	**Server platform:** Windows NT Server 3.51 or later, or AIX 4.2 or later **Client platform:** Windows 95, or Windows NT 4.0 TCP/IP

2.2 DB2 OLAP Server installation preparation

Before starting the DB2 OLAP Server for OS/390 installation, you need to collect specific information about your system environment. You can get this information from your DB2 database administrator or your system programmer.

The planning worksheet in Table 4 on page 26 will help you find all the information you will need.

DB2 OLAP Server for OS/390 can provide a relational (RSM) or multidimensional (MDSM) storage interface. Before installing the product, you must choose which interface you want to use. Refer to 1.8, “Multidimensional versus relational models” on page 15 before making your choice.

During the installation process, some tasks may have to be performed by other persons, depending on your organization:

- DASD space allocation
- SMP/E installation
- OS/390 UNIX setup
- Network communication setup
- OS/390 RACF user creation and security definitions
- System definitions (SYS1.PARMLIB, SYS1.PROCLIB access)
- DB2 for OS/390 setup (user authorization, resource allocation, DSNZPARM modification), only if you use the RSM interface

Using RSM, DB2 OLAP Server stores the application data into DB2 tablespaces. Fill out the RSM configuration column in Table 4 on page 26.

Using the multidimensional storage manager (MDSM), DB2 OLAP Server stores the data into flat files in the OS/390 UNIX environment. Fill out the MDSM configuration column in Table 4 on page 26.

Table 5 on page 28 shows the sample worksheet we used for the DB2 OLAP Server installation in the ITSO environment.

Table 4. DB2 OLAP Server for OS/390 installation planning worksheet

#	Information needed	RSM configuration	MDSM configuration
	Information needed for DB2 OLAP Server	-	-
1	HFS data set name for DB2 OLAP Server code		
2	HFS data set Storage Class name		
3	HFS data set Data Class name (1)		
4	HFS data set Management Class name (1)		
5	DB2 OLAP Server code directory		
6	DB2 OLAP Server API code directory		
7	DB2 OLAP Server user ID		
8	OS/390 Server UNIX TCP/IP address		
9	DB2 OLAP Server Started Procedure name		
	Information needed for DB2	-	-
10	DB2 location name		N/A
11	DB2 subsystem ID		N/A
12	DB2 OLAP Server DB2 database name		N/A
13	DB2 Stogroup name		N/A
14	DB2 Stogroup volume name		N/A
15	DB2 OLAP Server DB2 tablespace name		N/A
16	DB2 OLAP Server DB2 Admin tablespace name		N/A
17	DB2 Primary Qty for DB2 OLAP tablespace		N/A
18	DB2 Secondary Qty for DB2 OLAP tablespace		N/A

#	Information needed	RSM configuration	MDSM configuration
19	DB2 Primary Qty for DB2 OLAP Admin tablespace		N/A
20	DB2 Secondary Qty for DB2 OLAP Admin tablespace		N/A
21	DB2 SDSNLOAD library name		N/A
22	DB2 EXIT library name		N/A
	Information needed for RRS	-	-
23	OS/390 Sysplex name		N/A
24	Couple data set name		N/A
25	Couple data set volume name		N/A
26	RRS started procedure name		N/A
	License Information	-	-
27	How many additional concurrent user licenses did you purchase?		
28	Did you purchase the Partitioning Option?		
29	Did you purchase the Tools Bundle?		
30	Did you purchase the Objects Package?		
31	Did you purchase the Web Gateway?		
32	Did you purchase the Integration Server?		

Notes

1. The Data Class and the Management Class are optional.

2. During the MDSM installation, you will be prompted for the DB2 location name. If DB2 is not installed in your environment, enter *none*.

Table 5 gives you the sample installation worksheet we used at the ITSO.

Table 5. DB2 OLAP Server for OS/390 installation planning worksheet

#	Information needed	RSM configuration	MDSM configuration
	Information needed for DB2 OLAP Server	-	-
1	HFS data set name for DB2 OLAP Server code	OMVS.SC62.DB2OLAP.RSM	OMVS.SC62.DB2OLAP.MDSM
2	HFS data set Storage Class name	SCCOMP	SCCOMP
3	HFS data set Data Class name (1)	-	-
4	HFS data set Management Class name (1)	-	-
5	DB2 OLAP Server code directory	/u/essbase/rsm	/u/essbase/mdsm
6	DB2 OLAP Server API code directory	/u/essbase/rsm	/u/essbase/mdsm
7	DB2 OLAP Server user ID	OLAPR	OLAPM
8	OS/390 Server UNIX TCP/IP address	9.12.2.26	9.12.2.26
9	DB2 OLAP Server Started Procedure name	DB2OLAPR	DB2OLAPM
	Information needed for DB2	-	-
10	DB2 location name	DBH1	N/A
11	DB2 subsystem ID	DBH1	DBH1
12	DB2 OLAP Server DB2 database name	OLAPDB	N/A
13	DB2 Stogroup name	SGOLAP	N/A
14	DB2 Stogroup volume name	TOTDBK	N/A
15	DB2 OLAP Server DB2 tablespace name	OLAPTS	N/A
16	DB2 OLAP Server DB2 Admin tablespace name	ADMINTS	N/A
17	DB2 Primary Qty for DB2 OLAP tablespace	150000	N/A

#	Information needed	RSM configuration	MDSM configuration
18	DB2 Secondary Qty for DB2 OLAP tablespace	20000	N/A
19	DB2 Primary Qty for DB2 OLAP Admin tablespace	150000	N/A
20	DB2 Secondary Qty for DB2 OLAP Admin tablespace	20000	N/A
21	DB2 SDSNLOAD library name	DB2V6H1.SDSNLOAD	N/A
22	DB2 EXIT library name	DB2V6H1.SDSNEXIT	N/A
	Information needed for RRS	-	-
23	OS/390 Sysplex name	PLEX57	N/A
24	Couple data set name	SYS1.PLEX57.LOGR01	N/A
25	Couple data set volume name	PDGCD1	N/A
26	RRS started procedure name	RRS	N/A

2.3 Installation steps overview

This section provides a general overview of the steps to install DB2 OLAP Server with the MDSM or RSM interface.

- Install and customize OS/390 UNIX. Refer to Chapter 3, “Enabling OS/390 UNIX” on page 33 and 4.2.6, “Configure OS/390 UNIX” on page 64.
- Install and customize TCP/IP for OS/390 environment. Refer to 3.4, “TCP/IP considerations” on page 45.
- If you use RSM, install and customize RRS. Refer to 4.3.7, “Install and customize RRS” on page 79.
- If you use RSM, install and customize DB2 for OS/390 and refer to 4.3.8, “Configure DB2 for OS/390 for the DB2 OLAP Server” on page 85.
- Perform SMP/E installation and apply service levels. Refer to the DB2 OLAP Server program directory and 4.1, “SMP/E considerations” on page 53.
- Install the DB2 OLAP Server code. To install the code using the MDSM interface, refer to 4.2, “Post-SMP/E installation process for MDSM” on page 54. To install DB2 OLAP Server for OS/390 using the RSM interface, refer to 4.3, “Post-SMP/E installation process for RSM” on page 69.
- The steps for the Post-SMP/E installation process are:
  - Allocate an HFS data set for DB2 OLAP Server code
  - Create a new user ID for DB2 OLAP Server engine

- Unpack the tar file
- Run the setup script
- Update the environment settings for DB2 OLAP Server
- Configure the OS/390 UNIX environment
- Check and customize RRS (only with RSM)
- Configure DB2 for OS/390 for the DB2 OLAP Server (RSM only)
  - Check DB2 parameters in DSNZPARM
  - Install the DB2 Call Level Interface (CLI)
  - Perform the CLI test for the DB2 OLAP Server
- Create a database and tablespaces for DB2 OLAP Server (RSM only)
- Verify DB2 OLAP Server RACF password authorization
- Start the DB2 OLAP Server for OS/390
- Create a DB2 OLAP Server Started Task procedure

- Install the sample applications. Refer to 4.5, “Install the sample applications” on page 97.
- Install the Essbase Application Manager on a client workstation. Refer to 5.1, “Application Manager installation” on page 103.
- Install the Essbase Spreadsheet add-in features on a client workstation. Refer to 5.2, “Add-in feature installation” on page 106.
- Load the data into the sample application. Refer to *Essbase Installation Notes,* GC26-9237.

2.4 NLS considerations

DB2 OLAP Server supports various national languages. You will need to set some environment variables in the *.profile* file used by the DB2 OLAP Server user ID. These variables determine the settings for the locale, territory, and sort method for your country and language.

Currently, the following national languages are supported by DB2 OLAP Server:

Brazilian Portuguese
Chinese Simplified
Chinese Traditional
Czech, French
German
Hungarian
Italian
Japanese
Korean
Russian
Spanish

United Kingdom English
United States English

To customize the DB2 OLAP Server for your national language, refer to 4.2.5, “Update the environment settings for DB2 OLAP Server” on page 62.

Chapter 3. Enabling OS/390 UNIX

The DB2 OLAP Server for S/390 Version 1.1 runs on OS/390 UNIX. This chapter describes the basic, required components of the OS/390 UNIX for our installation:

- Hierarchical file system (HFS)
- UNIX interactive interface
- Customization of the OS/390 UNIX environment
- TCP/IP
- Security

For detailed information about OS/390 UNIX, refer to *OS/390 UNIX System Services Planning*, SC28-1890, and *OS/390 V2.6 UNIX System Services Implementation and Customization*, SG24-5178.

3.1 Hierarchical file system

The OS/390 UNIX file system is hierarchical and byte-oriented, so we call it the hierarchical file system (HFS). Finding a file in the file system is done by searching a directory or a series of directories. There is no concept of an OS/390 catalog that points directly to a file.

The hierarchical file system consists of:

- HFS files, which contain data or programs

  A file containing a load module, shell script, or REXX program is called an executable file. Files are stored in directories.

- Directories, which contain files, other directories, or both

  Directories are arranged hierarchically, in a structure that resembles an upside-down tree, with the root directory at the top and the branches at the bottom. The root is the first directory for the file system at the top of the tree and is designated by a slash (/). Figure 5 on page 35 is an example of a UNIX root directory, containing a typical set of subdirectories.

- Additional local or remote file systems, which are mounted on directories of the root file system

To the MVS system, the file hierarchy is a collection of HFS data sets. Each HFS data set is a mountable file system. The root file system is the first file system mounted. Subsequent file systems can be logically mounted on a

directory within the root file system or on a directory within any mounted file system.

Except for the direction of the slashes, the hierarchical file system is similar to a Disk Operating System (DOS), Windows, or an OS/2 file system.

When you install OS/390 UNIX, you define the root file system using the ROOT statement in the BPXPRMxx member of SYS1.PARMLIB. The root file system is the starting point for the overall HFS file structure. It contains the root directory and any related HFS files or subdirectories.

HFS data sets must be managed by SMS, and can span volumes with OS/390 Version 2.7 and DFSMS V1.5. They can grow to as many as 123 extents of DASD space, if secondary extents are specified in the allocation. JCL is typically used to create new HFS data sets by specifying DSNTYPE=HFS on the DD statement.

3.1.1 Recommended file system structure

OS/390 Version 2.6 or later delivers a single HFS containing the root directory and all OS/390 related products. This was done to simplify the installation and management of the OS/390 HFS structure.

Figure 5 on page 35 shows the single HFS containing a typical UNIX directory structure, with separate HFS data sets mounted at appropriate directory mount points. The number of actual directories varies depending on OS/390 version/release and product content.

At first, the OS/390 file system should be set up so that any post-install customization, or non-system data, is stored outside of (externalized) the OS/390 HFS. This will enable you to implement a new OS/390 HFS to replace the old, during an upgrade of the OS/390 operating system.

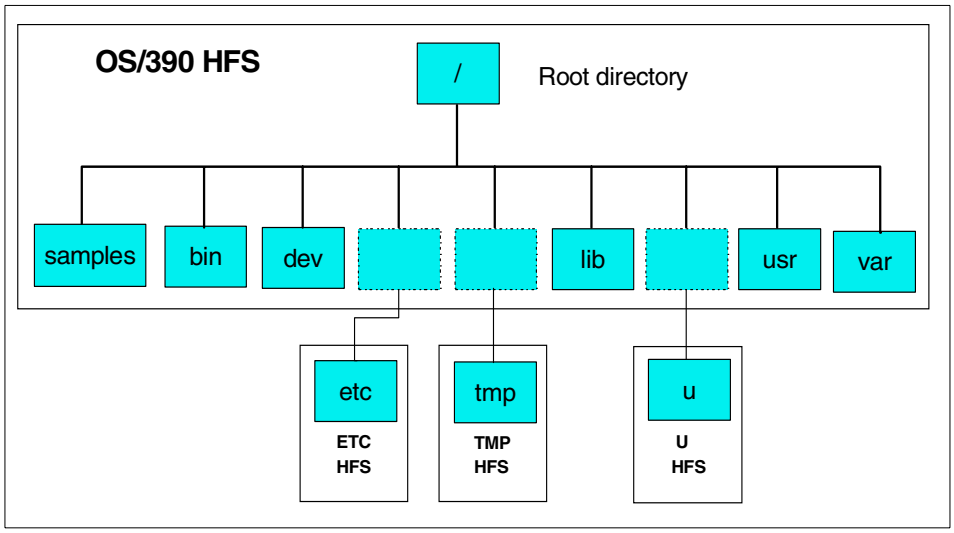


Figure 5. Example of an OS/390 HFS structure

The recommended method of externalizing customized or non-system data is to place it in one or more separate HFS data sets that are mounted at the appropriate directory mount points. To manage space, isolate failures, and reduce contention, users should have their own HFS data set, which can be sized for each user's needs. We recommend that you allocate an HFS data set for the DB2 OLAP Server, with separate HFS data sets for the OLAP applications.

At most installations, user home directories are named /u/<userid>, where <userid> is the user ID in lowercase characters. Separate mountable user file systems offer several advantages:

- They improve storage management because the system administrator need only allocate data sets large enough to accommodate the needs of individual users.
- They enable failure isolation because the system administrator can unmount the user file system that caused an error without affecting other users' data.
- They relieve the contention for system resources that could occur by having multiple users in a single file system.

3.1.2 MVS data sets and HFS

An HFS file data is byte-oriented, unlike most MVS data sets, which are record-oriented. Input and output (I/O) for HFS files is typically performed through the use of a data stream. Despite the differences between them, you can copy HFS files into MVS data sets, and MVS data sets into HFS files, using special TSO/E commands like OCOPY, OGET, and OPUT. You can access HFS files and manipulate data from application programs using defined C functions.

Figure 6 shows the difference between MVS data sets and a UNIX hierarchical file system.

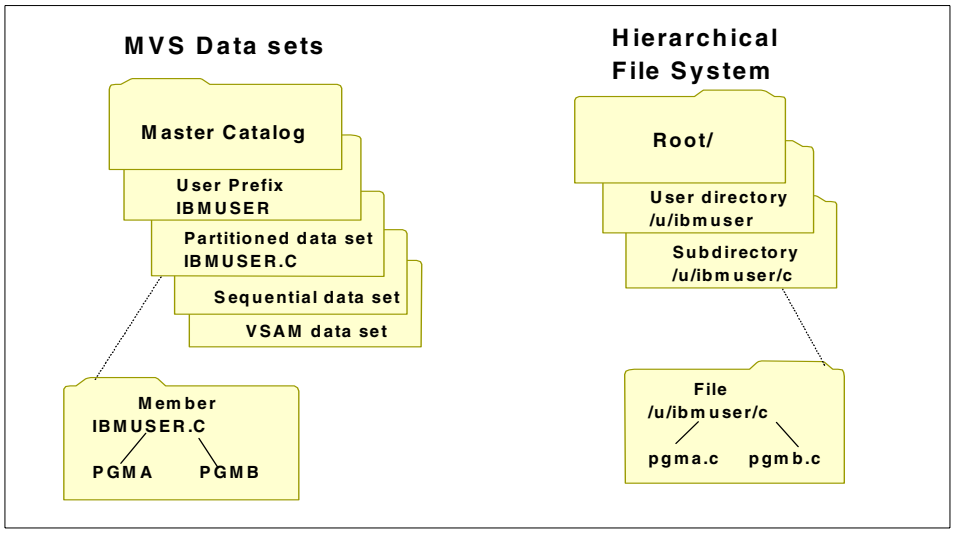


Figure 6. Comparison of MVS data sets and a hierarchical file system

The OS/390 master catalog is analogous to the root directory in a hierarchical file system.

The user prefix assigned to OS/390 data sets points to a user catalog. The user catalog is organized analogous to a user directory (/u/ibmuser) in the file system. Typically, one user owns all the data sets whose names begin with his/her user prefix. For example, the data sets belonging to the TSO/E user ID ibmuser all begin with the prefix ibmuser. There could be data sets named IBMUSER.C, and IBMUSER.C(PGMA).

In the file system, ibmuser would have a user directory named /u/ibmuser. Under the directory there could be subdirectories named /u/ibmuser/c and /u/ibmuser/c/pgma.

Of the various types of OS/390 data sets, a partitioned data set (PDS) is most like a user directory in the file system. In a partitioned data set such as IBMUSER.C, you could have members PGMA, PGMB, and so on. For example, you could have IBMUSER.C(PGMA) and IBMUSER.C(PGMB). A subdirectory such as /u/ibmuser/c can hold many files, such as pgma, pgmb, and so on.

3.1.3 Creating HFS data sets

When you install the software products in the OS/390 UNIX, you need to allocate separate HFS data sets for the products, and mount these separate HFS data sets onto the directory. These separate HFS data sets can be used to store data or programs unique to each product. This practice allows each product to use its own HFS data sets without impacting any other OS/390 UNIX users and programs. It is also a way of isolating each product or user for systems management purposes. We recommend that you adopt this approach to managing multiple file systems.

Figure 7 shows an example of how to create a new HFS data set.

```
//BI390C  JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=BI390C
//*************************************************************
//* SAMPLE JCL TO ALLOCATE AN MVS HFS DATASET
//*************************************************************
//STEP1  EXEC PGM=IEFBR14
//SYSPRINT  DD DSN=OMVS.SC62.DB2OLAP.MDSM,
//             DISP=(NEW,CATLG,DELETE),
//             SPACE=(CYL,(150,50,1)),
//             STORCLAS=SCCOMP,MGMTCLAS=STANDARD,
//             DSNTYPE=HFS
//*************************************************************
```

Figure 7. Sample JCL to allocate an HFS data set

Notes:

1. HFS data sets must be SMS managed. The storage class STORCLAS tells SMS what facilities this HFS data set requires. It may also be used by the SMS ACS routines to direct the data set to a particular storage group.
2. The DSNTYPE is HFS, which tells OS/390 that this is an HFS data set.

3.1.4 Creating the HFS file system for OLAP applications

You allocate the separate HFS for the application in exactly the same way as you create the user HFS. Choose a data set name that has the application name as one of the qualifiers, and a size that provides sufficient space for the application's requirements. As an application adds files and extends existing files, if more space is required, you can choose one of the following methods:

- Increase the size of the allocation.
- Create additional HFS data sets on different DASD volumes for an application and mount them at different mount points in the application's hierarchy.
- Allocate an HFS that will span DASD volumes with DFSMS V1.5.

3.1.5 Making HFS file systems available

After the OS/390 UNIX HFS data sets are allocated, you need to get them mounted at a mount point off the root directory to make them available. The preferred place to mount HFS data sets for a user is the /u mount point; however, you can place different mount points for the HFS data sets of the DB2 OLAP Server. There are two ways to accomplish this:

- Direct mount
- Automount facility

In this section, we describe direct mount only, which we used at the ITSO. If you need more information about the automount facility, refer to *OS/390 V2.6 UNIX System Services Implementation and Customization*, SG24-5178, or ask your OS/390 UNIX administrator.

Using direct mount

In order to mount a new HFS data set at the /u directory, your ID should have superuser authority. There are several ways to mount an HFS data set:

- Using the TSO MOUNT command:

  ```
  MOUNT FILESYSTEM('OMVS.SC62.DB2OLAP.RSM') TYPE(HFS)
  MOUNTPOINT('/u/essbase/rsm')
  ```

- Using the ISHELL File_systems pull-down

  Type in OMVS from ISPF Option 6 to enter the shell and execute the highlighted commands to mount the HFS data set OMVS.SC62.DB2OLAP.RSM.

- Adding an entry to the BPXPRMxx member of SYS1.PARMLIB as shown in Figure 8 for the automatic mount of the HFS data set upon each re-IPL.

```
MOUNT FILESYSTEM(OMVS.SC62.DB2OLAP.RSM)
TYPE(HFS)
MOUNTPOINT(/u/essbase/rsm)
MODE(RDWR)
```

Figure 8. A mounting entry of a HFS file in BPXPRMxx

3.1.6 Increasing the size of HFS data sets

Increasing the size of HFS data sets is simple because they can be unmounted and unallocated. To increase the size of the HFS data sets:

1. Unmount the HFS data set and any HFS data sets that are mounted at mount points lower in the tree. The data set will need to be unallocated before it can be unmounted. If an application, for example the DB2 OLAP Server, is accessing the HFS, it may need to be stopped until the dump and restore have been completed.
2. Use the JCL in Figure 9 to dump the HFS data set; in our example, OMVS.SC62.APPL.ACCTNG.

```
//DSSDUMP JOB (999,POK),'HFS DUMP',CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*
//* THIS JOB WILL CREATE NEW HFS AND COPY EXISTING HFS
//* INTO NEWLY CREATED HFS
//*
//SU       EXEC  PGM=ADRDSSU,REGION=6M
//SYSPRINT DD    SYSOUT=*
//HFSVOL   DD    UNIT=3390,VOL=SER=TOTDBK,DISP=SHR
//HFSOUT   DD    DSN=OMVS.SC62.APPL.SEQ,
//         DISP=(NEW,CATLG,DELETE),SPACE=(CYL,(100,50),RLSE),
//         UNIT=SYSALLDA,STORCLAS=SCCOMP
//SYSIN    DD *
 DUMP DATASET(INCLUDE(OMVS.SC62.APPL.ACCTNG)) -
      COMPRESS TOL(ENQF) -
      LOGINDDNAME(HFSVOL)  OUTDDNAME(HFSOUT) ALLDATA(*) ALLEXCP
/*
```

Figure 9. DUMP an HFS data set using DFDSS

3. Rename the HFS data set. Once the data set is no longer mounted, you can rename it from ISPF Option 3.4 or by using the R - Rename Data Set line command in OMVS.
4. Preallocate a larger HFS data set with the same name as the old smaller HFS data set. Refer to Figure 7 on page 37 for creating the new HFS data set.
5. Use the JCL in Figure 10 to restore the HFS data set with the REPLACE keyword.

```
//BI390A  JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*
//* RESTORE HFS DATA SET TO INCREASE THE SIZE
//*
//RESTORE  EXEC PGM=ADRDSSU
//SYSPRINT DD   SYSOUT=*
//HFSSEQ   DD   DSN=OMVS.SC62.APPL.SEQ,DISP=SHR
//HFSOUT   DD   UNIT=3390,VOL=SER=TSMS25,DISP=SHR
//SYSIN    DD *
 RESTORE  INDD(HFSSEQ) OUTDD(HFSOUT) TOL(ENQF) -
 DATASET(INCLUDE(OMVS.SC62.APPL.ACCTNG))       -
 STORCLAS(SCCOMP)          -
 REPLACE                   -
 CANCELERROR
/*
```

Figure 10. Restore HFS data set using DFDSS

3.2 OS/390 UNIX interface

A user can interact with OS/390 UNIX using the following interfaces:

- The application programming interface (API): It consists of C programming calls that can be used by C/370 programs to access OS/390 UNIX.
- The interactive interface: It is called the OS/390 UNIX shell. The shell is the command interpreter that accepts commands defined in the POSIX 1003.2 standard.

In this section we describe the OS/390 UNIX interactive interface.

Interactive users of OS/390 UNIX have a choice between using a UNIX-like interface (the UNIX shell), a TSO interface (TSO commands), or an ISPF interface (ISPF CUA dialog). Thus, users can choose the interface they are

most familiar with and get a quick start with OS/390 UNIX. Figure 11 shows two interfaces: OS/390 UNIX shell and ISHELL.

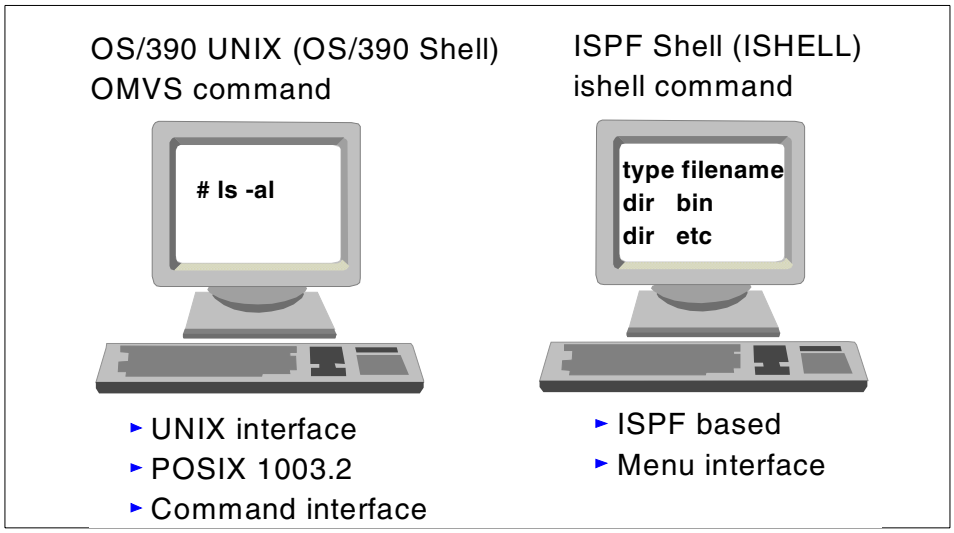


Figure 11. OS/390 UNIX interactive interface

There are some TSO commands that provide the interactive interfaces to OS/390 UNIX (shown in Figure 11), as follows:

- OMVS is an important TSO command that invokes the OS/390 UNIX shell.
- ISHELL invokes the ISPF shell. ISHELL is a good starting point for users familiar with TSO and ISPF that need to use OS/390 UNIX. This provides the panels for working with the hierarchical file system. There are also panels for mounting/unmounting file systems and for doing OS/390 UNIX administration.

3.2.1 OS/390 UNIX shell

The OS/390 UNIX shell is based on the UNIX system V shell and has some of the features from the UNIX Korn shell. The POSIX standard distinguishes between a command that is a directive to the shell to perform a specific task, and a utility that is the name of a program callable by name from the shell. To the users, there is no difference between a command and a utility.

The OS/390 UNIX shell provides the environment that has the most functions and capabilities. Shell commands can easily be combined in pipes or shell scripts and thereby become powerful new functions. A sequence of shell commands can be stored in a text file that can be executed. This is called a

shell script. The shell supports many of the features of a regular programming language.

3.2.2 Telnet

An OS/390 UNIX user can log in directly to the OS/390 UNIX shell using telnet. You can also use *rlogin* (from UNIX) or OCS (from RS/6000), but we describe telnet only, which we used during our test.

The telnet support comes with the TCP/IP OS/390 UNIX feature. It also uses the *inetd* daemon, which must be active and set up to recognize and receive the incoming telnet requests.

To invoke a telnet session from the DOS prompt:

1. Enter telnet
2. Click **Connect--> Remote system** in the option bar
3. Enter a hostname or IP address
4. Choose *telnet* or port number at the Port button
5. Choose *vt100* at the Term type button
6. Click **Connect**
7. After you enter the login ID and password (which are RACF authorized), you are connected to the OS/390 UNIX shell

This solution provides an asynchronous terminal interface to OS/390 UNIX. By default, the terminal interface is in line mode which is the same as the 3270 interface.

3.2.3 TSO commands

There are numerous shell commands you can use to create and work with directories and files. You can also use TSO/E commands to do certain tasks with the file system. Some of these are tasks that UNIX users traditionally perform within the shell.

ISHELL: Invoke the ISPF shell. This is a panel interface for performing many user and administrator tasks.

MKDIR: Create a directory. Unlike the `mkdir` shell command, this command does not create intermediate directories in a path name if they do not exist.

MOUNT: Add a mountable file system to the file hierarchy. To use this command, you must be a superuser.

OBROWSE: Browse (read but not update) an HFS file using the ISPF full-screen browse facility. It can also be used in the shell.

OCOPY: Copy an MVS data set member into an HFS file, using ddnames. Copy an HFS file into an MVS data set member, using ddnames. Copy an HFS file into another HFS file. Copy an MVS data set member into another MVS data set member.

OEDIT: Create or edit text using the ISPF editor from an OMVS session. It can also be used in the shell.

OGET: Copy an HFS file to an MVS sequential data set or partitioned data set member. You can specify text or binary data, and select code page conversion.

OGETX: Copy one or many files from a directory to a partitioned data set, a PDS/E, or a sequential data set. You can specify text or binary data, select code page conversion, allow a copy from lowercase filenames, and delete one or all suffixes from the filenames when they become PDS member names.

OPUT: Copy an MVS sequential data set or partitioned data set member to an HFS file. You can specify text or binary data, and select code page conversion.

OPUTX: Copy one or many members from a partitioned data set, PDS/E, or a sequential data set to a directory. You can specify text or binary data, select code page conversion, specify a copy to lowercase filenames, and append a suffix to the member names when they become filenames.

OSTEPLIB: Build a list of files that are sanctioned as valid step libraries for programs that have the set-user-ID or set-group-ID bit set. To use this command, you must be a superuser.

UNMOUNT (or UMOUNT): Remove a file system from the file hierarchy. To use this command, you must be a superuser.

3.3 Customize the OS/390 UNIX environment

In this section, we describe some parameters of OS/390 UNIX that are related to the DB2 OLAP Server for OS/390.

3.3.1 Customize the user profile

You need to customize the *.profile* of the DB2 OLAP Server supervisor or the user who will activate the DB2 OLAP Server. Refer to 4.2.5, “Update the environment settings for DB2 OLAP Server” on page 62 or 4.3.5, “Update the environment settings for DB2 OLAP Server” on page 77.

3.3.2 Changes to SYS1.PARMLIB

You need to set the following two OS/390 UNIX parameters to avoid possible errors when running the DB2 OLAP Server for OS/390.

- MAXASSIZE

  MAXASSIZE will be the region size for all processes created via *telnet*. This enables every process to allocate as much storage as is available, up to 2 GB. The default value is 40 MB.

  We recommend you to set this value to 2 147 483 647.

- MAXCPUTIME

  MAXCPUTIME indicates the CPU time that a process is allowed to use, in seconds. The default value for this parameter for OS/390 UNIX processes is set to 1000 seconds.

  Because the server processes for the DB2 OLAP Server for OS/390 can function as *daemon* processes, if MAXCPUTIME is not set to some much higher value, an OLAP transaction might fail.

  We recommend that you also set this value to 2 147 483 647. This provides a process with unlimited CPU time, which prevents a long-running OLAP process from being cancelled.

In order to check the currently assigned values of these parameters, go to the SDSF log panel and enter the MVS command:

```
D OMVS,O
```

To dynamically change these parameters, issue the following commands from the SDSF log panel:

```
SETOMVS MAXASSIZE=2147483647
SETOMVS MAXCPUTUME=2147483647
```

Change the SYS1.PARMLIB(BPXPRMxx) member to reflect these changes in your environment.

3.3.3 TSO region size of the user

If you are planning to run the DB2 OLAP Server from a TSO OMVS session (not from telnet), make sure to set the region size at least to 256 MB (or the maximum allowed at your company). If you use the default value, you might have errors (for example, running out of memory) during installation processes.

We recommend that you set the SIZE parameter to 256,000 (256 MB) when you create a new TSO user ID. See Figure 14 on page 50.

3.4 TCP/IP considerations

The eNetwork Communications Server IP provides networking support for OS/390 UNIX. You need to check the TCP/IP customizing information to run the DB2 OLAP Server for OS/390 in a client/server environment. Refer to *OS/390 eNetwork Communications Server: IP Configuration*, SC31-8513, and *TCP/IP Tutorial and Technical Overview*, SG24-5227, for a description of TCP/IP concepts and protocols.

Checking the hosts file

Check the OS/390 UNIX's IP address and hostnames at your installation. You can get this information in */etc/hosts*. In our environment, we have:

```
9.12.2.26 wtsc62oe wtsc62oe.itso.ibm.com
```

where 9.12.2.26 is the IP address, *wtsc62oe* is the hostname, and *wtsc62oe.itso.ibm.com* is the full domain name. When you connect to DB2 OLAP Server from Application Manager, Lotus 1-2-3, MS Excel, or any type of client described in Figure 2 on page 8, you should use either IP address or hostname.

TCP/IP port

DB2 OLAP Server uses TCP/IP port 1423. You should assign this port to be used by DB2 OLAP Server into the port assignment table, which is identified by a set of entries loaded through the //PROFILE DD statement in the TCP/IP start procedure. Figure 12 on page 46 shows an example of the table.

```
PORT
     20 TCP OMVS                ; OE FTP SERVER
     21 TCP OMVS                ; OE FTPD CONTROL PORT
     23 TCP OMVS                ; OE TELNET SERVER
     80 TCP OMVS                ; OE WEB SERVER
    111 UDP OMVS                ; OE PORTMAPPER SERVER
    111 TCP OMVS                ; OE PORTMAPPER SERVER
    443 TCP OMVS                ; OE WEB SERVER SSL PORT
    512 TCP OMVS                ; OE REMOTE EXECUTION SERVER
    513 TCP OMVS                ; OE RLOGIN SERVER
    514 TCP OMVS                ; OE REMOTE SHELL SERVER
    514 UDP OMVS                ; OE SYSLOGD SERVER
    515 TCP OMVS                ;
   1423 TCP OMVS                ; DB2 OLAP SERVER
```

Figure 12. Port assignment table

Ports coded in the table are reserved for use by the identifying address space or task name. You can assign port 1423 to OMVS for the DB2 OLAP Server by adding the port statement. If TCP/IP port 1423 is reserved by another address space, you will encounter an error like the following:

```
Unable to bind Host server socket on port [1423]
```

When you get this error, you can issue the *netstat* command to see what address space port 1423 is currently assigned to. Consult with your system support people for more details and proper action to correct the error.

TCP/IP affinity

You can start the DB2 OLAP Server process from a TSO OMVS or telnet session. If you have multiple TCP/IP stacks in your system and start the DB2 OLAP Server process from a TSO OMVS session with an incorrect TCP/IP stack, you might get a TCP/IP error like the following:

```
Network Error 112 Unable to Accept Connection
```

To avoid this, specify the correct TCP/IP stack by either:

- Using a //SYSTCPD DD statement in your TSO logon procedure, or
- Running the sample REXX procedure provided in Appendix B, “REXX procedure to create TCP/IP affinity” on page 227

This creates the right affinity between your TSO user ID and the TCP/IP stack.

3.5 Prepare for OS/390 UNIX security

As with all applications running on OS/390 UNIX, the DB2 OLAP Server for OS/390 is subject to its security constraints. In order to be able to install and run the DB2 OLAP Server, appropriate authority within OS/390 UNIX should be set up.

In this section we describe the OS/390 UNIX security considerations:

- OS/390 UNIX security
- Defining a user to RACF with UNIX authority
- HFS file authorization
- Authorization of the user who installs the DB2 OLAP Server for OS/390
- Authorization of the user who activates the DB2 OLAP Server

3.5.1 OS/390 UNIX security

In the OS/390 UNIX environment, as in other UNIX systems, every user on the system needs to be identified. For each user we define:

- User name
- Password (stored in encrypted form)
- User identifier (UID)
- Group identification number (GID)
- User's full name
- User's home directory
- File name of the shell program

The UID is a number between 0 and 2147483647. If two users are assigned the same UID, OS/390 UNIX views them as the same user, even if they have different user names and passwords. Two users with the same UID could read and write over each other's files and kill each other's processes. Therefore, you need to be careful in how you allocate UIDs.

The OS/390 UNIX environment, as with other UNIX systems, has the concept of groups. By using groups, you can connect many users who need to access a set of common files and directories.

OS/390 UNIX uses the same concept of user accounts as other UNIX systems, but it stores the information in a segment (the OMVS segment) of a

user profile in the RACF database. UNIX users are defined and managed with RACF commands.

3.5.2 Defining a user to RACF with UNIX authority

How do you define a RACF user ID with UNIX authority? The user IDs for installing the server and for running the server both need OS/390 UNIX authorization. The user can be either a TSO user ID with OS/390 UNIX access (OMVS segment) or only an OS/390 UNIX user. You can run a batch job to accomplish this or use ISPF panels to define users dynamically.

We show two batch jobs, that define a new RACF user with a UNIX user ID only (Figure 13), and a new RACF user with a TSO account and a UNIX user ID (Figure 14 on page 50).

```
//BI390C  JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=BI390C
//*************************************************************
//* SAMPLE JCL TO CREATE A NEW RACF USER
//* WITH ONLY OMVS ACCESS
//*************************************************************
//IKJEFT   EXEC PGM=IKJEFT01,REGION=3072K
//SYSUADS  DD DSN=SYS1.UADS,DISP=SHR
//SYSLBC   DD DSN=SYS1.BRODCAST,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
  ADDUSER OLAPR PASSWORD(OLAP390) +
    NAME('OLAP User') +
    OWNER(RCONWAY) UACC(NONE) DFLTGRP(SYS1) +
    OMVS(UID(334) HOME(/u/olapr) PROGRAM(/bin/sh))
```

Figure 13. Defining a new UNIX user to RACF

In Figure 13, the user ID is OLAPR, and the password is OLAP390.

When defining the user, you need to add OMVS information by defining an OMVS segment in RACF; see the bold line.

1. The user ID that *installs* the DB2 OLAP Server must have a UID=0 in order to be able to update the root directory. The user ID that *runs* the DB2 OLAP server should not have a UID=0, and the number you specify will depend on your installation. In our example, we give the user a UID of 334.

2. Enter an initial directory path name for this user. This is the directory that the users are placed in when they first log on. It is known as the HOME path. In our example, it is /u/olapr.

   Note: The home directory must exist for the server to use UNIX services, if you do not use the Automount facility. (With Automount, a home directory for a user is automatically created.) Create this directory (from a superuser) using the command `mkdir /u/olapr` (this assumes that the /u directory exists). You should then change ownership of the directory to the user using the `chown` command (see 3.5.5, “Authorization of the user who activates the DB2 OLAP Server” on page 52).

3. Enter a program path name for this user. This should always be /bin/sh - the shell program.

For those who will use OS/390 UNIX through OMVS, define new RACF users with TSO account information (see Figure 14 on page 50). We recommend that you assign the user region size as 256 MB. For detailed information, ask your TSO administrator.

```
//BI390C  JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=BI390C
//*************************************************************
//* SAMPLE JCL TO CREATE A NEW RACF USER FOR DB2 OLAP SERVER
//* WITH TSO AND OMVS ACCESS
//*************************************************************
//IKJEFT   EXEC PGM=IKJEFT01,REGION=3072K
//SYSUADS  DD DSN=SYS1.UADS,DISP=SHR
//SYSLBC   DD DSN=SYS1.BRODCAST,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
  ADDUSER OLAPR PASSWORD(OLAP390) +
    NAME('OLAP User') +
    OWNER(RCONWAY) UACC(NONE) DFLTGRP(SYS1) +
    AUTHORITY(USE) GRPACC +
    TSO(ACCTNUM(ACCNT#) PROC(IKJACCNT) SIZE(256000) MAXSIZE(0) +
    UNIT(SYSALLDA)) +
    OMVS(UID(334) HOME(/u/essbase/rsm) PROGRAM(/bin/sh))
  PERMIT ACCNT# CLASS(ACCTNUM) ID(OLAPR)
  PERMIT IKJACCNT CLASS(TSOPROC) ID(OLAPR)
  PERMIT JCL CLASS(TSOAUTH) ID(OLAPR)
  PERMIT RECOVER CLASS(TSOAUTH) ID(OLAPR)
  PERMIT OPER CLASS(TSOAUTH) ID(OLAPR)
  ALU OLAPR PASSWORD(OLAP390) NOEXPIRE
  ADDSD 'OLAPR.*' UACC(NONE) OWNER(OLAPR)
  LISTDSD DATASET('OLAPR.*')
  SETROPTS REFRESH RACLIST(TSOPROC ACCTNUM TSOAUTH)
  DEFINE ALIAS (NAME('OLAPR') RELATE('CATALOG.TOTICF2.VTOTCAT')) +
         CATALOG('MCAT.OS3R7V02.VTOTCAT')
/*
```

Figure 14. Defining a new TSO and UNIX user to RACF

SIZE is the region size for all processes created via OMVS session. This enables every process to allocate the storage up to 256 MB.

3.5.3 HFS file authorization

The security information for each file in the OS/390 UNIX HFS is stored with the file. If we list the files in an HFS directory using the command `ls -al`, we will see information similar to Figure 15 on page 51.

```
HAHN:/u/essbase/rsm: >ls -al
drwxr-xr-x   6 OLAPR  SYS1        8192 Dec  8 17:03 .
drwxr-xr-x   4 OMER   SYS1        8192 Jan  4 16:00 ..
-rw-r--r--   1 OLAPR  SYS1        1079 Nov 16 09:31 .profile
-rw-------   1 OLAPR  SYS1         698 Dec 16 14:42 .sh_history
-rwx------   1 OLAPR  SYS1           0 Dec  7 11:57 BPXBATCH.ERR
-rwx------   1 OLAPR  SYS1         273 Dec  7 11:57 BPXBATCH.OUT
-rw-r--r--   1 OLAPR  SYS1           4 Nov 16 09:39 Essbase.id
-rw-r--r--   1 OMER   SYS1       74790 Dec 20 14:55 Essbase.log
-rw-r--r--   1 OMER   SYS1          17 Dec  8 17:03 SYSINFO.OUT
drwxr-xr-x   6 OLAPR  SYS1        8192 Oct  1 13:25 api
drwxr-xr-x   2 OLAPR  SYS1        8192 Dec  2 18:20 app
-rw-r-----   1 OLAPR  SYS1          28 Nov 16 09:06 arbor.id
-rw-r-----   1 OLAPR  SYS1         444 Nov 16 09:06 arborenv.doc
drwxr-xr-x   2 OLAPR  SYS1        8192 Dec 20 14:54 bin
drwxr-xr-x   2 OLAPR  SYS1       16384 Oct  1 13:24 locale
-rwxr-xr-x   1 OLAPR  SYS1          42 Nov 16 11:58 start.sh
```

Figure 15. HFS file permission bits

The last field is the file name.

The third field OLAPR is the owner of the file and the fourth field SYS1 is the default group to which OLAPR belongs. These are as defined in RACF.

The first field is the file's permission bits. They define the access levels that users have to the file:

- Character one defines the file:
  - d means it is a directory or container
  - '-' means it is a non-directory file
- Characters two to four define the access right that the file owner has:
  - r means that read access is allowed
  - w means that write access is allowed
  - x means that execute access is allowed
- Characters five to seven define the access right that anyone in the file owner's group has, in the same way as for the file owner.
- Characters eight to ten define the access right that all other users have, in the same way as for the file owner.

In Figure 15 on page 51, the file *start.sh* can only be written to by the owner, but everyone on the system can read it.

3.5.4 Authorization of the user who installs the DB2 OLAP Server

The user ID that is used to install the DB2 OLAP Server for OS/390 must be defined to RACF with a valid OMVS segment (a valid UID and GID), and must have a UID of 0 (superuser) to be able to mount the HFS data sets.

The user may either be defined with a permanent UID of 0 or be permitted read access to the BPX.SUPERUSER profile in the FACILITY class, thus enabling the use of the `su` command.

3.5.5 Authorization of the user who activates the DB2 OLAP Server

The DB2 OLAP Server runs as a user on OS/390. The user ID is the one that is used to start the server and must be defined to RACF with a valid OMVS segment (a valid UID and a valid GID). The user ID must have authority to access all of the DB2 OLAP Server files on the HFS, including:

- The server's home directory (/u/essbase/rsm, in our case)
- All the files under the OLAP server's home directory
- The dsnaoini file
- The rsm.cfg file if you choose the relational storage manager (RSM)

We recommend that you do not run the DB2 OLAP Server from a user ID with a UID of 0. The server runs agents on behalf of users, and it is possible that a user could therefore run an agent with superuser authority.

You can change the owning user ID and group ID for a file by using the UNIX command `chown`. For example, in the DB2 OLAP Server's home directory, issue the following commands to change the owner name:

```
chown -R ownername *
chown -R ownername .*
chown -R ownername .
```

Chapter 4. DB2 OLAP Server installation

The DB2 OLAP Server allows storing of data either using the multidimensional storage manager (MDSM) or the relational storage manager (RSM) interface. Because both of these interfaces cannot run at the same time, you must decide which of these interfaces you want to use before starting the installation process. If you need to switch storage managers after installation, reinstall the DB2 OLAP Server code in another directory. If you don't, your existing files will be lost.

This chapter focuses on the following:

- SMP/E considerations
- The installation process for MDSM
- The installation process for RSM
- The sample application installation

4.1 SMP/E considerations

Before installing DB2 OLAP Server for OS/390, you should review the current Preventive Service Planning (PSP) information.

You can install DB2 OLAP Server for OS/390 using either of the following:

- MVS Custom-Built Product Delivery Offering (CBPDO)
- Stand-alone product tape

To install DB2 OLAP Server for OS/390, follow the SMP/E installation process. The SMP/E dialogs may be used to accomplish the SMP/E installation steps. The RECEIVE step generates the sample jobs to assist you in installing DB2 OLAP Server for OS/390.

Do the following to install the DB2 OLAP Server code:

1. Unload the sample JCL from the product tape
2. Update and execute the allocation jobs:
   a. Allocate the SMP/E target and distribution libraries
   b. Allocate the HFS paths for DB2 OLAP Server
3. Create DDDEF entries for SMP/E target and distribution libraries
4. Perform SMP/E RECEIVE
5. Perform SMP/E APPLY CHECK
6. Perform SMP/E APPLY
7. Perform SMP/E ACCEPT CHECK

8. Perform SMP/E ACCEPT

For detailed information about SMP/E steps, refer to *Program Directory for DB2 OLAP Server for OS/390*, GI10-8198.

The TSO user ID used to perform the SMP/E installation must be UID=0 or be permitted to the BPX.SUPERUSER profile in the FACILITY class to access the /etc and /usr/lpp directory.

If you run DB2 OLAP Server from a TSO OMVS session, the REGION parameter on the logon panel should be set to 256000.

Note

The SMP/E installation creates the tar file *DB2OLAP.Z* in a new OS/390 UNIX directory */usr/lpp/essbase/tar.* This is the default directory name used by the SMP/E process. You can decide to choose another directory name, then change all occurrences of this name in the installation jobs accordingly.

At this point, if you will use the multidimensional storage manager interface, go to 4.2, “Post-SMP/E installation process for MDSM” on page 54. If you will use the relational storage manager interface, go to 4.3, “Post-SMP/E installation process for RSM” on page 69.

You will get many directories and files during the installation process. For more information, refer to Appendix C, “Summary of DB2 OLAP Server files” on page 229.

4.2 Post-SMP/E installation process for MDSM

Use this procedure if you choose to install DB2 OLAP Server using the multidimensional storage manager (MDSM) interface.

This post-installation process is intended to expand the tar file created previously by the SMP/E process and set up the DB2 OLAP Server configuration and environment.

To perform the Post-SMP/E installation process, use a TSO user ID with the following authorities:

- Appropriate RACF security authority, according to your environment, to perform the installation tasks defined in this chapter, such as creating an

HFS data set, creating a RACF user ID, setting up system parameters, and creating an STC procedure

- Superuser access to the OS/390 UNIX (UID=0), or permission to the BPX.SUPERUSER profile in the FACILITY class to store files in the /etc/essbase directory
- Read access permission to the CLASS(FACILITY) BPX.FILEATTR.PROGCTL profile to be able to set the program controlled extended attribute bit on some executable files

To check your OS/390 UNIX UID, issue the following command:

- In the TSO environment (*username* is your own user ID):

  ```
  TSO LU username OMVS
  ```

- In the OS/390 UNIX environment (OMVS):

  ```
  id
  ```

4.2.1 Prepare an HFS data set for DB2 OLAP Server code

We recommend that you install the DB2 OLAP Server code for MDSM in a separate HFS MVS data set.

You can prepare an HFS data set as follows:

1. Create an HFS data set to store the DB2 OLAP Server code (see Figure 16).

```
//BI390C  JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=BI390C
//*************************************************************
//* SAMPLE JCL TO ALLOCATE AN MVS HFS DATASET USED TO STORE
//* DB2 OLAP SERVER CODE
//*************************************************************
//STEP1  EXEC PGM=IEFBR14
//SYSPRINT  DD DSN=OMVS.SC62.DB2OLAP.MDSM,
//             DISP=(NEW,CATLG,DELETE),
//             SPACE=(CYL,(350,50,1)),
//             STORCLAS=SCCOMP,MGMTCLAS=STANDARD,
//             DSNTYPE=HFS
//*************************************************************
```

Figure 16. Sample JCL to allocate the DB2 OLAP Server HFS data set

2. Create the directory in which you want to install the DB2 OLAP Server code. We called it the DB2 OLAP Server home directory (/u/essbase/mdsm). You can do this using one of the following:

- ISPF shell (ish) or TSO command processor

```
mkdir '/u/essbase' MODE(7 5 5)
mkdir '/u/essbase/mdsm' MODE(7 5 5)
```

- UNIX command from the UNIX shell (OMVS)

```
mkdir -p -m 755 /u/essbase/mdsm
```

3. Mount this new HFS data set to the directory you just created. Use the TSO ISPF Shell interface (TSO `ISH` command) or run the job shown in Figure 17.

```
//BI390C  JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=BI390C
//*************************************************************
//* SAMPLE JCL TO MOUNT THE MVS HFS DATA SET CREATED TO STORE
//* DB2 OLAP SERVER CODE TO /u/essbase/mdsm/ DIRECTORY
//*************************************************************
//STEP1  EXEC PGM=IKJEFT01
//SYSPRINT  DD SYSOUT=*
//SYSTSPRT  DD SYSOUT=*
//SYSTSIN   DD *
   MOUNT FILESYSTEM('OMVS.SC62.DB2OLAP.MDSM') -
         MOUNTPOINT('/u/essbase/mdsm') TYPE(HFS) MODE(RDWR)
//*************************************************************
```

Figure 17. Sample JCL to mount the HFS data set to the MDSM directory

Ensure the right permission bits (755) are set up for the /u/essbase/mdsm directory. You can change the permission bits by using the TSO ISPF Shell or by issuing the following UNIX command:

```
chmod 755 /u/essbase/mdsm
```

To automatically mount the MVS HFS data set on each IPL, you can add the following definition in the SYS1.PARMLIB(BPXPRMxx) member of your installation:

```
MKDIR '/u/essbase/mdsm' MODE(7 5 5)
MOUNT FILESYSTEM('OMVS.SC62.DB2OLAP.MDSM') TYPE(HFS) +
MODE(RDWR) MOUNTPOINT('/u/essbase/mdsm')
```

Figure 18. BPXPRMxx to mount the HFS data set to the MDSM directory

4.2.2 Create a supervisor ID for DB2 OLAP Server

We recommend that you create a new RACF user ID that will be used as a supervisor ID when the DB2 OLAP Server engine starts. This user ID does not need to be a TSO user ID, but must be defined as an OS/390 UNIX (OMVS) user ID (see a sample supervisor ID definition on Figure 19).

This supervisor ID does not need to be a superuser (UID<>0).

```
//BI390C  JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=BI390C
//*************************************************************
//* SAMPLE JCL TO CREATE A NEW RACF USER FOR DB2 OLAP SERVER
//* WITH ONLY OMVS ACCESS
//*************************************************************
//IKJEFT   EXEC PGM=IKJEFT01,REGION=3072K
//SYSUADS  DD DSN=SYS1.UADS,DISP=SHR
//SYSLBC   DD DSN=SYS1.BRODCAST,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
  ADDUSER OLAPM PASSWORD(OLAP390) +
    NAME('MOLAP Server User') +
    OWNER(RCONWAY) UACC(NONE) DFLTGRP(SYS1) +
    OMVS(UID(334) HOME(/u/essbase/mdsm) PROGRAM(/bin/sh))
  ALU OLAPM PASSWORD(OLAP390) NOEXPIRE
/*
```

Figure 19. Sample JCL to create an OMVS user ID for MDSM

Note

To avoid having to change the password regularly, according to your security policy, you can use the NOEXPIRE parameter of the `ALU` RACF command.

4.2.3 Unpack the tar file

From the directory /usr/lpp/essbase/tar, expand the product tar file by entering the following command:

```
tar xpfo DB2OLAP.Z
```

After the tar file is successfully expanded, two new directories (390 and docs) and one file (setup.sh) are created in /usr/lpp/essbase/tar, as shown in Figure 20 on page 58.

```
BI390C:/usr/lpp/essbase/tar: >ls -al
total 39904
drwxrwx---   4 STC      SYS1        8192 Jan 24 16:52 .
drwxr-xr-x  43 STC      SYS1        8192 Feb  7 15:52 ..
drwxr-xr-x   4 STC      SYS1        8192 Jan 24 16:52 390
-rw-r--r--   1 STC      SYS1    20377600 Feb  7 16:16 DB2OLAP.Z
drwxrwx---   2 STC      SYS1        8192 Jan 24 15:08 docs
-r-xr-xr-x   1 STC      SYS1        2595 Jan 24 15:23 setup.sh
BI390C:/usr/lpp/essbase/tar: >
```

Figure 20. The files and directories created in the DB2 OLAP Server tar directory

4.2.4 Run the setup script

From the same directory, now run the setup.sh script to start the installation process. Answer the questions shown in the following example. Before this step, you should have filled in the planning worksheet provided in Table 4 on page 26. This worksheet will help you answer the questions during the installation process.

The installation dialog is divided into three parts:

1. The server installation dialog
2. The license enablement dialog
3. The API product component installation dialog

Server installation dialog
After entering the `setup.sh` command, you are prompted to enter the following information needed to install the DB2 OLAP Server code. Refer to the sample dialog in Figure 21 on page 60:

- Accept the terms of the Restricted Rights for Server: **yes**
- Select a Data Storage Option: **M**
- Enter your DB2 OLAP Server directory: **/u/essbase/mdsm**
- Enter your DB2 Subsystem ID: **DBH1** (or none if DB2 is not installed)

Note

When you install the DB2 OLAP Server MDSM interface, you are prompted to specify a DB2 Subsystem ID name. If you do not have any DB2 subsystem installed in your environment, just reply *none*. If you do, indicate the DB2 subsystem ID corresponding to your environment.

DB2 OLAP Server uses this information to create a dsnaoini file in the /etc/essbase directory. It offers SQL interface functions such as SQL Drill-Through, and the ability to load the multidimensional data from a DB2 relational database.

License Enablement dialog

You now enter the License Enablement dialog that allows DB2 OLAP Server to use the features you have purchased for use on this server. You need to refer to your purchasing documents to complete the following dialog:

- Did you purchase any additional concurrent user licenses?
- Enter the number of additional concurrent user licenses you purchased:
- Did you purchase the Partitioning Option?
- Did you purchase the Tools Bundle?
- Did you purchase the Objects Package?
- Did you purchase the Web Gateway?
- Did you purchase the Integration Server?

All your responses will be summarized and written in the LICENSE.LOG file in the DB2 OLAP Server home /bin directory.

If you purchase additional options (or want to remove options) after the DB2 OLAP Server installation, you can modify the options registration by running the LICENSE program located in the /bin directory of the DB2 OLAP Server home directory.

API product component installation dialog

Then you are prompted to enter the following information needed to install the DB2 OLAP Server API code. The API code is not mandatory, but we strongly recommend that you perform the installation now.

Enter your DB2 OLAP Server directory: /u/essbase/mdsm.

The sample shown in Figure 21 shows our answers to the questions.

```
BI390C:/usr/lpp/essbase/tar: >setup.sh
IBM Software Corporation
All Rights Reserved
Copyright (c) 2000

IBM DB2 OLAP Server 1.1 for OS/390 (SERVER) - February 2000
Licensed Material - Property of IBM
Copyright (c) 2000 IBM. All right reserved.
US Government Users Restricted Rights -
Use, duplication or disclosure restricted
by GSA ADP Schedule Contract with IBM Corp.

You must agree to the terms of the IBM
International Program License Agreement
(IPLA) before installing or using this Program.
The IPLA is included in your Program package.

Accept the terms of the Restricted Rights Legend described
above and continue with the install? (yes/no): yes

Select a Data Storage Option:
Enter "R" to install the Relational Database option.
Enter "M" to install the Multidimensional Data Store option.
Enter your choice (R, M or C to Cancel): M

Please enter your DB2 OLAP Server directory (/u/essbase/):
/u/essbase/mdsm
Installing DB2 OLAP Server, please wait...
Please enter your DB2 Subsystem ID: DBH1
```

Figure 21. Sample setup dialog for MDSM (1 of 3)

```
DB2 OLAP Server License Enablement
This program enables DB2 OLAP Server to use the features you
have purchased for use on this server. It also creates a
record of the features that are assigned to this server.
To complete this step, you must have access to the purchasing
or packing documents that show which features you purchased.
No changes are made until you answer all questions.

The edition of the server you purchased allows access by 1 user.
Did you purchase any additional concurrent user licenses? (Y/N) Y
Enter the number of additional concurrent user licenses you purchased.
10
Did you purchase the Partitioning Option? (Y/N) Y
Did you purchase the Tools Bundle? (Y/N) Y
Did you purchase the Objects package? (Y/N) Y
Did you purchase the Web Gateway? (Y/N) Y
Did you purchase the Integration Server? (Y/N) Y
Your responses are summarized below.
Quantity Description
       1 Standard Edition    1.1
      10 Additional Concurrent Users
       1 Partitioning Option 1.1
       1 Tools Bundle        1.1
       1 Objects             1.1
       1 Web Gateway         1.1
       1 Integration Server  1.1

Are you satisfied with this allocation of options to this server? Y
Essbase Registration Utility - 5.0.2 Patch2 (03/19/99)
Copyright(c) 1991-1999 Hyperion Solutions Corporation
U.S. Patent Number 5,359,724
All rights reserved

Writing serial number 180000050032639A-00BF203D63A
Writing license number into ESSBASE
Writing license number into ESSSVR
Your server was successfully updated.
SERVER installation finished successfully.
Please refer to file /u/essbase/mdsm/essbaseenv.doc
and set up your environment accordingly.
```

Figure 22. Sample setup dialog for MDSM (2 of 3)

```
The API product component will be installed next.

Please enter your DB2 OLAP Server directory (/u/essbase/):
/u/essbase/mdsm
Installing DB2 OLAP Server API, please wait...
API installation finished successfully.

Please refer to file /u/essbase/mdsm/essbaseenv.doc
and set up your environment accordingly.
setup.sh (top-level): done
BI390C:/usr/lpp/essbase/tar: >
```

Figure 23. Sample setup dialog for MDSM (3 of 3)

4.2.5 Update the environment settings for DB2 OLAP Server

At this point, depending on the directory names you choose, you must have the following directories (the names are those we chose in our environment):

- Directory that contains the tar file and the installation script and directories: /usr/lpp/essbase/tar
- DB2 OLAP Server directory that contains the DB2 OLAP Server and API code: /u/essbase/mdsm
- Directory that contains the dsnaoini configuration file: /etc/essbase

You now have to update the environment for DB2 OLAP Server:

1. Change the owner of DB2 OLAP Server directory.

   All the directories and files in the DB2 OLAP Server home directory should be owned by the DB2 OLAP Server supervisor ID (OLAPM in our environment). From the DB2 OLAP Server home directory (/u/esbase/mdsm), issue the following OMVS commands to change the owner name (you can also use the TSO ISPF shell as well):

   ```
   chown -R OLAPM *
   chown -R OLAPM .*
   chown -R OLAPM .
   ```

2. Update the DB2 OLAP Server supervisor ID profile.

   The installation program does not update the UNIX profile file for the supervisor ID. Create a .profile file in the DB2 OLAP Server home directory (/u/essbase/mdsm in our environment) using the essbaseenv.doc

file information located in the DB2 OLAP Server home directory. A sample file is shown in Figure 24:

```
export ARBORPATH=/u/essbase/mdsm
export LIBPATH=$LIBPATH:/u/essbase/mdsm/bin
export PATH=$PATH:/u/essbase/mdsm/bin
export DSNAOINI=/etc/essbase/dsnaoini
export _CEE_RUNOPTS='HEAPPOOLS(ON)'
export _BPX_SHAREAS='NO'
export LC_ALL="En_US.IBM-1047"
export GC_LANG="English_UnitedStates.IBM1047"
export DB2OLAP_CLIENT_CODESET="ISO8859-1"
```

Figure 24. Sample .profile file for the DB2 OLAP Server supervisor ID (MDSM)

The `export` command sets these variables when the profile is executed. Check the following points:

- The ARBORPATH variable must be set with your DB2 OLAP Server code home directory (/u/essbase/mdsm in our example).
- Use the _BPX_SHAREAS='NO' parameter. This variable is used by the spawn callable service of the OS/390 UNIX kernel. NO specifies that the child process is to be created in a new address space. This is the value required for DB2 OLAP Server.
- The DSNAOINI variable has to be set with the path and file name of the dsnaoini file, which is the initialization file of the DB2 ODBC functions. The default is /etc/essbase/dsnaoini.
- Set up the language environment run-time options:

  The Application Manager client for DB2 OLAP Server for OS/390 runs on a Windows ASCII platform, while the server runs on OS/390, which is an EBCDIC environment. Code set conversion must take place for the two components to work together.

  DB2 OLAP Server for OS/390 supports multiple national languages. Set up the environment variables according to the national language you use. In our environment, we used the United States English national language as shown in Figure 36. For other language environments, refer to Appendix A, "National language environment variables" on page 225.

Note

OS/390 UNIX does not support file names that contain double-byte characters. Therefore, application and database names cannot contain double-byte characters.

4.2.6 Configure OS/390 UNIX

Before starting the DB2 OLAP Server, you need to check the values of two OS/390 UNIX parameters. We recommend that you assign the maximum value to these two parameters:

- MAXASSIZE = 2 147 483 647
- MAXCPUTIME = 2 147 483 647

In order to check the currently assigned values of these parameters, go to the SDSF log panel and enter the command:

```
D OMVS,O
```

To dynamically change these parameters, issue the following commands from the SDSF log panel:

```
SETOMVS MAXASSIZE=2147483647
SETOMVS MAXCPUTUME=2147483647
```

To automatically use these values on each successive IPL, you need to modify the SYS1.PARMLIB(BPXPRMxx) member to reflect these changes. Refer to 4.2.6, “Configure OS/390 UNIX” on page 64.

4.2.7 Verify RACF password authorization for DB2 OLAP Server

You need to verify first that the `p e`xtended attribute is set for the PWDTEST and ESSBASE modules, located in the DB2 OLAP Server /bin directory (/u/essbase/mdsm/bin). From the OMVS environment, issue the commands:

```
ls -lE ESSBASE
ls -lE PWDTEST
```

You should get the result shown in Figure 25 on page 65.

```
BI390C:/u/essbase/mdsm/bin: >ls -lE ESSBASE
-rwxr-xr-x  -ps  1 STC      SYS1     1994752 Feb  7 16:25 ESSBASE
BI390C:/u/essbase/mdsm/bin: >ls -lE PWDTEST
-rwxr-xr-x  -ps  1 STC      SYS1       77824 Jan 24 15:57 PWDTEST
BI390C:/u/essbase/mdsm/bin: >
```

Figure 25. List of file attributes for DB2 OLAP Server modules

If the `p` attribute is not set correctly, you must use the `extattr` command to set it:

```
extattr +p ESSBASE
extattr +p PWDTEST
```

To modify the program controlled extended attributes, your TSO user ID must be defined with read permission to the CLASS(FACILITY) BPX.FILEATTR.PROGCTL profile.

You now need to test whether the user ID you use to log in to the DB2 OLAP Server is defined to RACF:

1. Execute the profile file you created previously in the DB2 OLAP Server home directory (/u/essbase/mdsm):

   . .profile

2. Run the PWDTEST program to test the DB2 OLAP Server security options. Enter the user ID and password you have previously created for DB2 OLAP Server. Refer to Figure 26 on page 66 for a PWDTEST sample execution.

   You should get a message indicating the test was successful.

```
BI390C:/u/essbase/mdsm/bin: >PWDTEST

This program prompts for a RACF user ID and password. It then checks
that OS/390 UNIX is configured to interact with RACF using
the user ID and password provided.

Enter user ID:
OLAPM

Enter password:
OLAP390

The test was successful. OS/390 UNIX is correctly configured
to interact with RACF and the user ID and password are specified to
RACF.

BI390C:/u/essbase/mdsm/bin: >
```

Figure 26. Sample execution for the PWDTEST program for MDSM

4.2.8 Configure DB2 for DB2 OLAP Server to use the SQL interface

This step is optional and applies only if DB2 for OS/390 is installed in your environment, and you want to access your DB2 subsystem to load the multidimensional cube using the DB2 OLAP Server interface from the Application Manager.

You need to perform the following steps (refer to 4.3.7, “Install and customize RRS” on page 79, 4.3.8, “Configure DB2 for OS/390 for the DB2 OLAP Server” on page 85, and 4.3.9, “Perform the CLI test” on page 88):

1. Install and customize RRS
2. Check DB2 parameters in DSNZPARM
3. Enable the DB2 Call Level Interface (CLI)
4. Set the CLI initialization file
5. Perform the CLI test

Note

Be sure you are using the same code pages and national language features in DB2 for OS/390, DB2 OLAP Server, and DB2 OLAP Server client applications.

4.2.9 Start DB2 OLAP Server for OS/390

You are now ready to start DB2 OLAP Server for OS/390.

There are three different ways to start the server:

- In a foreground mode: This mode is required for the first start. Use OS/390 UNIX from a TSO session or from a telnet session to start the DB2 OLAP Server.
- In a background mode: Use OS/390 UNIX from a TSO session or from a telnet session to start the DB2 OLAP Server. The session is prompted after the server has started.
- Using an MVS started task procedure: This is recommended to run the DB2 OLAP Server in an operational environment. To set up an MVS started task procedure, refer to 6.1, “Activating the DB2 OLAP Server” on page 143.

To start DB2 OLAP Server for OS/390 do the following:

1. Log on to OS/390 UNIX from a TSO session or a telnet session. Make sure you run the .profile file created during the installation phase and stored in the DB2 OLAP Server home directory (/u/essbase/mdsm).
2. Change to the DB2 OLAP Server bin directory (/u/essbase/mdsm/bin).
3. Start the DB2 OLAP Server in the foreground by entering:

   ```
   ESSBASE
   ```

4. You are prompted to enter:
   - Your company name
   - The DB2 OLAP Server supervisor ID and password
5. Confirm the information you entered.

   After this initialization step, you should receive the message:

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

   Figure 27 on page 68 shows a sample dialog when starting the server *for the first time* in the foreground mode.

```
BI390C:/u/essbase/mdsm/bin: >ESSBASE

Code page in use: IBM1047
11 login system
Hyperion Essbase OLAP Server - 5.0.2 Patch2 (03/19/99)
Copyright 1991-1998 Hyperion Solutions Corporation.
US Patent Number 5,359,724
All Rights Reserved.
Serial number:  180000050032639A-00BF203D63A
Welcome to the Hyperion Essbase OLAP Server.

Enter your company name Ý¨: IBM
Enter your name Ý¨: OLAPM
Enter the system password Ý¨: OLAP390

You have entered the following details:
Company name:     IBM
Your name:        OLAPM
System password:  OLAP390

These will be used to create the initial system security information
including the system supervisor.
Are these details correct? (y/n): Y

Waiting for Client Requests...
```

Figure 27. Sample dialog when starting DB2 OLAP Server with MDSM

6. You can now stop the DB2 OLAP Server and choose to start it in background mode.
   - To stop the DB2 OLAP Server, enter the command:

     `EXIT`
   - To start the DB2 OLAP Server in the background, enter the command:

     `ESSBASE password -b&`
   - To stop a DB2 OLAP Server started in background mode, use the ESSCMD `SHUTDOWNSERVER` command (Refer to 6.2, “Deactivating the DB2 OLAP Server” on page 147).

Now go to 4.5, “Install the sample applications” on page 97 to set up sample applications. Then, go to Chapter 5, “Setting up clients” on page 103, to install the client workstation options, such as Application Manager and Spreadsheet Add-ins.

4.3 Post-SMP/E installation process for RSM

Use this procedure if you choose to install DB2 OLAP Server using the RSM interface. DB2 for OS/390 Version 5 Release 1 or Version 6 Release 1 should already be installed on your system.

This post-installation process is intended to expand the tar file created previously by the SMP/E process and set up the DB2 OALP Server configuration and environment.

To perform the post-SMP/E installation process, use a TSO user ID with the following authorities:

- Appropriate RACF security authority, according to your environment, to perform the installation tasks defined in this chapter such as creating an HFS data set, creating a RACF user ID, setting up system parameters, and creating an STC procedure
- Superuser access to the OS/390 UNIX (UID=0), or permission to the BPX.SUPERUSER profile in the FACILITY class to store files in the */etc/essbase* directory
- Read access permission to the CLASS(FACILITY) BPX.FILEATTR.PROGCTL profile to be able to set the program controlled extended attribute bit on some executable files
- SYSADM DB2 authority

To check your OS/390 UNIX UID, issue the following command:

In the TSO environment (*username* is your own user ID):

```
TSO LU username OMVS
```

In the OS/390 UNIX environment (OMVS):

```
id
```

4.3.1 Prepare an HFS data set for DB2 OLAP Server code

We recommend that you install the DB2 OLAP Server code for RSM in a separate HFS MVS data set.

You can prepare an HFS data set as follows:

1. Create an HFS data set to store the DB2 OLAP Server code (see Figure 28 on page 70).

```
//BI390B JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=BI390B
//*************************************************************
//* SAMPLE JCL TO ALLOCATE AN HFS MVS DATASET USED TO STORE
//* DB2 OLAP SERVER CODE
//*************************************************************
//STEP1  EXEC PGM=IEFBR14
//SYSPRINT  DD DSN=OMVS.SC62.DB2OLAP.RSM,
//             DISP=(NEW,CATLG,DELETE),
//             SPACE=(CYL,(350,50,1)),
//             STORCLAS=SCCOMP,MGMTCLAS=STANDARD,
//             DSNTYPE=HFS
//*************************************************************
```

Figure 28. Sample JCL to allocate the DB2 OLAP Server HFS data set

2. Create the directory in which you want to install the DB2 OLAP Server code. We will call it the DB2 OLAP Server home directory (/u/essbase/rsm in our environment). You can do this using one of the following:
   - ISPF Shell (ISH) or TSO command processor

     ```
     mkdir '/u/essbase' MODE(7 5 5)
     mkdir '/u/essbase/rsm' MODE(7 5 5)
     ```

   - UNIX command from the UNIX shell (OMVS)

     ```
     mkdir -p -m 755 /u/essbase/rsm
     ```

3. Mount this new HFS data set to the directory you just created. Use the TSO ISPF Shell interface (TSO `ISH` command) or run the job shown in Figure 29.

```
//BI390B JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=BI390B
//*************************************************************
//* SAMPLE JCL TO MOUNT THE HFS MVS DATA SET CREATED TO STORE
//* DB2 OLAP SERVER CODE TO /u/essbase/rsm/ DIRECTORY
//*************************************************************
//STEP1  EXEC PGM=IKJEFT01
//SYSPRINT  DD SYSOUT=*
//SYSTSPRT  DD SYSOUT=*
//SYSTSIN   DD *
   MOUNT FILESYSTEM('OMVS.SC62.DB2OLAP.RSM') -
         MOUNTPOINT('/u/essbase/rsm') TYPE(HFS) MODE(RDWR)
//*************************************************************
```

Figure 29. Sample JCL to mount the HFS data set to the RSM directory

Ensure that the right permission bits (755) are set up for the /u/essbase/rsm directory. You can change the permission bits by using the ISPF Shell or by issuing the following UNIX command:

```
chmod 755 /u/essbase/rsm
```

To automatically mount the MVS HFS data set on each IPL, you can add the following definition in the SYS1.PARMLIB(BPXPRMxx) member of your installation.

```
MKDIR '/u/essbase/rsm' MODE(7 5 5)
MOUNT FILESYSTEM('OMVS.SC62.DB2OLAP.RSM') TYPE(HFS) +
MODE(RDWR) MOUNTPOINT('/u/essbase/rsm')
```

Figure 30. BPXPRMxx to mount the HFS data set to the RSM home directory

4.3.2 Create a supervisor ID for DB2 OLAP Server

We recommend that you create a new RACF user ID that will be used as a supervisor ID when the DB2 OLAP Server engine starts. This user does not need to be a TSO user ID, but must be defined as an OS/390 UNIX (OMVS) user ID (see a sample supervisor ID definition in Figure 31).

This supervisor ID does not need to be a superuser (UID<>0).

```
//BI390B JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=BI390B
//*************************************************************
//* SAMPLE JCL TO CREATE A NEW RACF USER FOR DB2 OLAP SERVER
//* WITH ONLY OMVS ACCESS
//*************************************************************
//IKJEFT   EXEC PGM=IKJEFT01,REGION=3072K
//SYSUADS  DD DSN=SYS1.UADS,DISP=SHR
//SYSLBC   DD DSN=SYS1.BRODCAST,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
  ADDUSER OLAPR PASSWORD(OLAP390) +
    NAME('ROLAP Server User') +
    OWNER(RCONWAY) UACC(NONE) DFLTGRP(SYS1) +
    OMVS(UID(335) HOME(/u/essbase/rsm) PROGRAM(/bin/sh))
  ALU OLAPR PASSWORD(OLAP390) NOEXPIRE
/*
```

Figure 31. Sample JCL to create an OMVS user ID for RSM

Note

To avoid having to change the password regularly, according to your security policy, you can use the NOEXPIRE parameter of the `ALU` RACF command.

4.3.3 Unpack the tar file

From the directory /usr/lpp/essbase/tar, expand the product tar file by entering the following command:

```
tar xpfo DB2OLAP.Z
```

After the tar file is successfully expanded, two new directories (390 and docs) and one file (setup.sh) are created in /usr/lpp/essbase/tar, as shown in Figure 32 on page 72.

```
BI390B:/usr/lpp/essbase/tar: >ls -al
total 39904
drwxrwx---   4 STC      SYS1        8192 Jan 24 16:52 .
drwxr-xr-x  43 STC      SYS1        8192 Feb  7 15:52 ..
drwxr-xr-x   4 STC      SYS1        8192 Jan 24 16:52 390
-rw-r--r--   1 STC      SYS1    20377600 Feb  7 16:16 DB2OLAP.Z
drwxrwx---   2 STC      SYS1        8192 Jan 24 15:08 docs
-r-xr-xr-x   1 STC      SYS1        2595 Jan 24 15:23 setup.sh
BI390B:/usr/lpp/essbase/tar: >
```

Figure 32. The files and directories created in the DB2 OLAP Server tar directory

4.3.4 Run the setup script

From the same directory, now run the setup.sh script to start the installation process. Answer the questions as shown in the following example. Before this step, you should have filled out the planning worksheet provided in Table 4 on page 26. This worksheet will help you answer the questions during the installation process.

The installation dialog is divided into three parts:

1. The server installation dialog
2. The license enablement dialog
3. The API product component installation dialog

Server installation dialog
After entering the `setup.sh` command, you are prompted to enter the following information needed to install the DB2 OLAP Server code. Refer to the sample dialog in Figure 33 on page 75:

- Accept the terms of the Restricted Rights for Server: **yes**
- Select a Data Storage Option: **R**
- Enter your DB2 OLAP Server directory: **/u/essbase/rsm**
- Enter your database location name: **DBH1**
- Enter your database tablespace: **OLAPDB.OLAPTS**
- Enter the tablespace name for the administration tables: **OLAPDB.ADMINTS**
- Enter your DB2 Subsystem ID: **DBH1**

License Enablement dialog
You now enter the License Enablement dialog that allows DB2 OLAP Server to use the features you have purchased for use on this server. You need to refer to your purchasing documents to complete the following dialog:

- Did you purchase any additional concurrent user licenses?
- Enter the number of additional concurrent users licenses you purchased:
- Did you purchase the Partitioning Option?
- Did you purchase the Tools Bundle?
- Did you purchase the Objects Package?
- Did you purchase the Web Gateway?
- Did you purchase the Integration Server?

All your responses will be summarized and written in the LICENSE.LOG file in the /bin directory.

If you purchase additional options (or want to remove options) after the DB2 OLAP Server installation, you can modify the options registration by running the LICENSE program located in the /bin directory of the DB2 OLAP Server home directory.

API product component installation
Then you are prompted to enter the following information needed to install the DB2 OLAP Server API code. The API code is not mandatory, but we strongly recommend you perform the installation now.

Enter your DB2 OLAP Server directory: /u/essbase/rsm.

Notes

The DB2 database and tablespaces will be created later in the installation process. Refer to 4.3.10, “Create a database and tablespaces for DB2 OLAP Server” on page 89.

DB2 OLAP Server uses this information to create the dsnaoini file and the rsm.cfg configuration file in the /etc/essbase directory. You can modify the parameters later by editing these two files.

The sample log in Figure 33 on page 75 gives you an idea of what dialog you should have. Our answers to the questions have been highlighted.

```
BI390B:/usr/lpp/essbase/tar: >setup.sh
IBM Software Corporation
All Rights Reserved

Copyright (c) 2000
IBM DB2 OLAP Server 1.1 for OS/390 (SERVER) - February 2000
Licensed Material - Property of IBM
Copyright (c) 2000 IBM. All right reserved.
US Government Users Restricted Rights -
Use, duplication or disclosure restricted
by GSA ADP Schedule Contract with IBM Corp.
You must agree to the terms of the IBM
International Program License Agreement
(IPLA) before installing or using this Program.
The IPLA is included in your Program package.

Accept the terms of the Restricted Rights Legend described
above and continue with the install? (yes/no): yes
Select a Data Storage Option:
Enter your choice (R, M or C to Cancel): R

Please enter your DB2 OLAP Server directory (/u/essbase/):
/u/essbase/rsm
Installing DB2 OLAP Server, please wait...
Enter your database location name ?
DBH1
Enter your database tablespace (database_name.tablespace_name).
OLAPDB.OLAPTS
Enter the tablespace name for the administration tables
in the following form (database_name.admin_tablespace_name).
This must be a 32k page tablespace.
OLAPDB.ADMINTS
Please enter your DB2 Subsystem ID: DBH1
```

Figure 33. Sample setup dialog for RSM (1 of 3)

```
DB2 OLAP Server License Enablement
This program enables DB2 OLAP Server to use the features you
have purchased for use on this server.  It also creates a
record of the features that are assigned to this server.
To complete this step, you must have access to the purchasing
or packing documents that show which features you purchased.
No changes are made until you answer all questions.

The edition of the server you purchased allows access by 1 user.
Did you purchase any additional concurrent user licenses? (Y/N) Y
Enter the number of additional concurrent user licenses you purchased.
10
Did you purchase the Partitioning Option? (Y/N) Y
Did you purchase the Tools Bundle? (Y/N) Y
Did you purchase the Objects package? (Y/N) Y
Did you purchase the Web Gateway? (Y/N) Y
Did you purchase the Integration Server? (Y/N) Y

Your responses are summarized below.
Quantity Description
       1 Standard Edition    1.1
      10 Additional Concurrent Users
       1 Partitioning Option 1.1
       1 Tools Bundle        1.1
       1 Objects             1.1
       1 Web Gateway         1.1
       1 Integration Server  1.1
Are you satisfied with this allocation of options to this server? (Y/N):
Y
Essbase Registration Utility - 5.0.2 Patch2 (03/19/99)
Copyright(c) 1991-1999 Hyperion Solutions Corporation
U.S. Patent Number 5,359,724
All rights reserved
Writing serial number 180000050032639A-00BF203D63A
Writing license number into ESSBASE
Writing license number into ESSSVR
Your server was successfully updated.
SERVER installation finished successfully.
Please refer to file /u/essbase/rsm/essbaseenv.doc
and set up your environment accordingly.
```

Figure 34. Sample setup dialog for RSM (2 of 3)

```
The API product component will be installed next.

Please enter your DB2 OLAP Server directory (/u/essbase/):
/u/essbase/rsm
Installing DB2 OLAP Server API, please wait...

API installation finished successfully.

Please refer to file /u/essbase/rsm/essbaseenv.doc
and set up your environment accordingly.

setup.sh (top-level): done
BI390B:/usr/lpp/essbase/tar: >
```

Figure 35. Sample setup dialog for RSM (3 of 3)

4.3.5 Update the environment settings for DB2 OLAP Server

At this point, depending on the directory names you choose, you must have the following directories (the names are those we choose in our environment):

- Directory that contains the tar file and the installation script and directories: /usr/lpp/essbase/tar
- DB2 OLAP Server directory that contains the DB2 OLAP Server and API code: /u/essbase/rsm
- Directory that contains the dsnaoini and the rsm.cfg configuration files: /etc/essbase

You now have to update the environment for DB2 OLAP Server:

1. Change the owner of the DB2 OLAP Server directory.

   All the directories and files in the DB2 OLAP Server home directory should be owned by the DB2 OLAP Server supervisor ID (OLAPR in our environment). From the DB2 OLAP Server home directory (/u/esbase/rsm), issue the following OMVS commands to change the owner name (you can also use the TSO ISPF shell):

   ```
   chown -R OLAPR *
   chown -R OLAPR .*
   chown -R OLAPR .
   ```

2. Update the DB2 OLAP Server supervisor ID profile.

   The installation program does not update the UNIX profile file for the supervisor ID. Create a .profile file in the DB2 OLAP Server home

directory (/u/essbase/rsm in our environment) using the essbaseenv.doc file information located in the DB2 OLAP Server home directory. A sample file is shown in Figure 36.

```
export STEPLIB=DB2V61H1.SDSNEXIT:DSN610.SDSNLOAD:CEE.SCEERUN
export ARBORPATH=/u/essbase/rsm
export LIBPATH=$LIBPATH:/u/essbase/rsm/bin
export PATH=$PATH:/u/essbase/rsm/bin
export DSNAOINI=/etc/essbase/dsnaoini
export _CEE_RUNOPTS='HEAPPOOLS(ON)'
export _BPX_SHAREAS='NO'
export LC_ALL="En_US.IBM-1047"
export GC_LANG="English_UnitedStates.IBM1047"
export DB2OLAP_CLIENT_CODESET="ISO8859-1"
```

Figure 36. Sample .profile file for the DB2 OLAP Server supervisor ID (RSM)

The `export` command sets these variables when the profile is executed. Check the following points:

- The ARBORPATH variable must be set with your DB2 OLAP Server code home directory (/u/essbase/rsm in our example).
- Use the _BPX_SHAREAS='NO' parameter. This variable is used by the spawn callable service of the OS/390 UNIX kernel. NO specifies that the child process is to be created in a new address space. This is the value required for DB2 OLAP Server.
- The DSNAOINI variable has to be set with the path and file name of the dsnaoini file, which is the initialization file of the DB2 ODBC functions. The default is /etc/essbase/dsnaoini.
- Set up the language environment run-time options:

  The Application Manager client for DB2 OLAP Server for OS/390 runs on a Windows ASCII platform, while the server runs on OS/390, which is an EBCDIC environment. Code set conversion must take place for the two components to work together.

  DB2 OLAP Server for OS/390 supports multiple national languages. Set up the environment variables according to the national language you use. In our environment, we used the United States English national language as shown in Figure 36. For other language environments, refer to Appendix A, "National language environment variables" on page 225.

Note

OS/390 UNIX does not support file names that contain double-byte characters. Therefore, application and database names cannot contain double-byte characters.

4.3.6 Configure OS/390 UNIX

Before starting the DB2 OLAP Server, you need to check the value of two OS/390 UNIX parameters. We recommend that you assign the maximum value to these two parameters:

- MAXASSIZE = 2 147 483 647
- MAXCPUTIME = 2 147 483 647

In order to check the currently assigned values of these parameters, go to the SDSF log panel and enter the command:

```
D OMVS,O
```

To dynamically change these parameters, issue the following commands from the SDSF log panel:

```
SETOMVS MAXASSIZE=2147483647
SETOMVS MAXCPUTUME=2147483647
```

To automatically use these values on each successive IPL, you need to modify the SYS1.PARMLIB(BPXPRMxx) member to reflect these changes. Refer to 4.2.6, “Configure OS/390 UNIX” on page 64.

4.3.7 Install and customize RRS

DB2 OLAP Server for OS/390, using the RSM interface needs the OS/390 Resource Recovery Services (RRS) component to be activated and to work with DB2 for OS/390.

For more details on this section, refer to *MVS Setting Up a Sysplex*, GC28-1779, or ask your S/390 system administrator.

This section introduces RRS and explains how to implement it.

4.3.7.1 Introduction to RRS

Resource recovery consists of protocols and programming interfaces that allow an application program to make consistent changes to multiple protected resources managed by multiple resource managers.

OS/390 includes services that when requested, can coordinate changes to one or more protected resources managed by a resource manager (for example DB2). The Resource Recovery Services (RRS) ensure that all changes are made or none of them are made.

RRS is the sync-point manager; it uses a two-phase commit protocol to coordinate changes to protected resources. The DB2 exit routine that gets control in response to RRS takes actions that commit or back out changes to a resource it manages.

DB2, as a resource manager, controls and manages access to a resource. It provides an API that allows the application program to read and change a protected resource.

The DB2 OLAP Server engine accesses the protected resources and requests changes to them.

RRS should be active on an OS/390 system that has programs involved in resource recovery. In a sysplex, RRS should be active on every OS/390 image that might take part in distributed resource recovery.

RRS uses five log streams, defined in a couple data set, that are shared by the systems in a sysplex. Every OS/390 image with RRS running needs access to a coupling facility and the DASD on which the system logger log streams are defined.

The five RRS logs are:

- RRS archive log (optional)
- RRS resource manager data log
- RRS main unit of recovery (UR) state log
- RRS delayed UR state log
- RRS restart log

The OS/390 system logger component manages log streams based on the policy information in the LOGR couple data set. A primary and an alternate couple data set are recommended.

4.3.7.2 Configure RRS

To set up RRS, do the following:

1. Ensure that your system logger is active, because RRS requires the system logger.
2. Set up the log streams.
3. Establish the priority of the RRS address space.

4. Define RRS as a subsystem.
5. Start the RRS subsystem.

Define the couple data set and activate the OS/390 system logger
Whether you are running in a sysplex or in a monoplex mode, you have to define a couple data set and activate your system logger, as follows:

1. Check if the system logger is already active by issuing the following command from the SDSF panel:

   ```
   D LOGGER,CONN
   ```

2. If the system logger is not active, define the LOGR couple data sets as shown in Figure 37. The IXCL1DSU utility formats the LOGR, which is a couple data set.

```
//BI390B  JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=BI390B
//*************************************************************
//* SAMPLE JOB TO DEFINE LOGR COUPLE DATASET
//*************************************************************
//STEP1    EXEC PGM=IXCL1DSU
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
     DEFINEDS SYSPLEX(PLEX57)
              MAXSYSTEM(2)
              DSN(SYS1.PLEX57.LOGR01) VOLSER(PDGCD1)
              CATALOG
         DATA TYPE(LOGR)
              ITEM NAME(LSR) NUMBER(100)
              ITEM NAME(LSTRR) NUMBER(60)
              ITEM NAME(DSEXTENT) NUMBER(10)
/*
```

Figure 37. Sample JCL to define an RRS LOGR couple data set

3. Modify the SYS1.PARMLIB(COUPLExx) member to add the couple data set definitions (see Figure 38 on page 82).

```
/*********************************************/
/* SAMPLE SYS1.PARMLIB COUPLExx MEMBER       */
/*********************************************/
 COUPLE SYSPLEX(&SYSPLEX.)
        PCOUPLE(SYS1.XCF.CDS02)
        ACOUPLE(SYS1.XCF.CDS03)
        CLEANUP(30)
        RETRY(10)

 DATA TYPE(LOGR)
        PCOUPLE(SYS1.XCF.LOGR00)
        ACOUPLE(SYS1.XCF.LOGR01)
```

Figure 38. Sample couple data set definition

To activate the system logger, IPL the system or issue the following `SETXCF` commands to bring the LOGR couple data sets online without IPLing the system:

```
SETXCF COUPLE,TYPE=LOGR,PCOUPLE=(primary_couple_data_set)
SETXCF COUPLE,TYPE=LOGR,ACOUPLE=(alternate_couple_data_set)
```

Set up the RRS log streams

1. Ensure that the system logger is already active with the command:

   ```
   D LOGGER,CONN
   ```

2. Define the log streams for RRS using the sample JCL in Figure 39 on page 83. The IXCMIAPU utility adds, updates, lists, or deletes policy data on the LOGR couple data set.

```
//BI390B  JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=BI390B
//*************************************************************
//* SAMPLE JCL TO DEFINE LOGSTREAMS FOR RRS
//*************************************************************
//LOGRPOL  EXEC PGM=IXCMIAPU
//STEPLIB  DD   DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD   SYSOUT=*
//SYSABEND DD   SYSOUT=*
//SYSIN    DD   *
     DATA TYPE(LOGR) REPORT(YES)
         DELETE LOGSTREAM NAME(IXGLOGR.PLEX57.ARCHIVE)
         DELETE LOGSTREAM NAME(IXGLOGR.PLEX57.RM.DATA)
         DELETE LOGSTREAM NAME(IXGLOGR.PLEX57.MAIN.UR)
         DELETE LOGSTREAM NAME(IXGLOGR.PLEX57.DELAYED.UR)
         DELETE LOGSTREAM NAME(IXGLOGR.PLEX57.RESTART)
         DEFINE LOGSTREAM
           NAME(ATR.PLEX57.ARCHIVE) DASDONLY(YES) HLQ(LOGR)
           MODEL(NO) LS_SIZE(1024) STG_SIZE(1024)
           LOWOFFLOAD(0) HIGHOFFLOAD(80)
           RETPD(15) AUTODELETE(YES)
         DEFINE LOGSTREAM
           NAME(ATR.PLEX57.RM.DATA) DASDONLY(YES) HLQ(LOGR)
           MODEL(NO) LS_SIZE(1024) STG_SIZE(1024)
           LOWOFFLOAD(0) HIGHOFFLOAD(80)
           RETPD(15) AUTODELETE(YES)
         DEFINE LOGSTREAM
           NAME(ATR.PLEX57.MAIN.UR) DASDONLY(YES) HLQ(LOGR)
           MODEL(NO) LS_SIZE(1024) STG_SIZE(1024)
           LOWOFFLOAD(0) HIGHOFFLOAD(80)
           RETPD(15) AUTODELETE(YES)
         DEFINE LOGSTREAM
           NAME(ATR.PLEX57.DELAYED.UR) DASDONLY(YES) HLQ(LOGR)
           MODEL(NO) LS_SIZE(1024) STG_SIZE(1024)
           LOWOFFLOAD(0) HIGHOFFLOAD(80)
           RETPD(15) AUTODELETE(YES)
         DEFINE LOGSTREAM
           NAME(ATR.PLEX57.RESTART) DASDONLY(YES) HLQ(LOGR)
           MODEL(NO) LS_SIZE(1024) STG_SIZE(1024)
           LOWOFFLOAD(0) HIGHOFFLOAD(80)
           RETPD(15) AUTODELETE(YES)
```

Figure 39. Sample JCL to define RRS log streams

Note

In the previous example, the RRS log streams are defined on DASD only. If you plan to use RRS across a sysplex environment, you must define the RRS log streams in the coupling facility. Review the parameters accordingly.

Establish the priority of the RRS address space

The priority of RRS should be higher than that of other resource managers, such as DB2, CICS, and IMS, and lower than that of JES and VTAM.

Define the RRS subsystem

1. Create the RRS started task procedure shown in Figure 40 in SYS1.PROCLIB(RRS). A sample job can be found in SYS1.SAMPLIB(ATRRRS).

```
//RRS      PROC GNAME='',CTMEM=''
//*********************************************************************
//* SAMPLE RRS STARTED TASK PROCEDURE
//*********************************************************************
//*
//* o GNAME=rrsgroupname
//* o CTMEM=ctracemembername
//*
//*  Examples of valid parameter strings:
//*
//*      PARM='GNAME=PLEX1 CTMEM=CTIRRS00'
//*      PARM='CTMEM=CTIRRS00      GNAME=PLEX1'
//*      PARM='GNAME=PLEX1    '
//*      PARM='         CTMEM=CTIRRS00             '
//*
//*********************************************************************
//RRS      EXEC PGM=ATRIMIKE,REGION=0M,TIME=NOLIMIT,
//             PARM='GNAME=&GNAME CTMEM=&CTMEM'
//
```

Figure 40. Sample RRS started task procedure

2. Modify the SYS1.PARMLIB(IEFSSNxx) member by adding the following definition:

   ```
   SUBSYS SUBNAME(RRS)
   ```

3. Issue the `SETSSI` command to define the new subsystem:

```
SETSSI ADD,SUBNAME=RRS
```

Start RRS

- Start RRS by entering the following command from SYSLOG:

  ```
  START RRS,SUB=MSTR
  ```

- If you need to stop RRS, you can use the command:

  ```
  SETATRRS CANCEL or FORCE RRS,ARM
  ```

Automatically activate RRS

To activate RRS automatically at each IPL, add the `START RRS` command in the SYS1.PARMLIB(COMMNDxx) member:

```
COM='START RRS'
```

4.3.8 Configure DB2 for OS/390 for the DB2 OLAP Server

Before starting DB2 OLAP Server, you need to perform these steps to configure the DB2 environment on your system. No connection is made to DB2 until you start DB2 OLAP Server and create your first OLAP application.

Notes

Check the PTF level needed on DB2 for OS/390 before customizing the DB2 environment for the RSM interface. Refer to 2.1.2, "S/390 Server software requirements" on page 20.

Be sure you are using the same code pages and national language features in DB2 for OS/390, DB2 OLAP Server, and DB2 OLAP Server client applications.

4.3.8.1 Check DB2 parameters in DSNZPARM

Make sure that the following two parameters are set as shown in your DSNZPARM:

- CACHEDYN=YES
- MAXKEEPD=5000

If not, you must modify these two parameters in the job DSNTIJUZ in the HLQ.NEW.SDSNSAMP data set. Stop the DB2 subsystem, run this job, and restart DB2.

4.3.8.2 Enable the DB2 Call Level Interface (CLI)

DB2 Call Level Interface (CLI) is IBM's callable SQL interface used by the DB2 family of products. DB2 OLAP Server uses function-calls to pass dynamic SQL statements as function arguments. DB2 CLI is based on the Microsoft Open Database Connectivity (ODBC) specification, and the X/Open Call Level Interface specification. On the DB2 for OS/390 platform, DB2 CLI support is implemented as a CLI/ODBC driver that is loaded at run-time into the application address space.

DB2 OLAP Server for OS/390 uses CLI to communicate with DB2 for OS/390. You have to set up the CLI environment on your platform and make sure the CLI works with DB2 OLAP Server for OS/390.

To enable the DB2 the Call Level Interface:

1. Make sure ODBC modules are restored in your DB2 library. ODBC modules are shipped on separate tapes from the DB2 base code.
2. Bind the ODBC module DSNACLI. The sample DSNTIJCL job is located in your DSN610.SDSNSAMP library. You can also use the sample JCL in Figure 41 on page 87. You can get a return code equal to 4 (warning).

For more detailed information, refer to *ODBC Guide and Reference Version 6*, SC26-9005. With DB2 Version 5, refer to *Call Level Interface Guide and Reference*, SC26-8959.

```
//BI390B  JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=BI390B
/*JOBPARM SYSAFF=SC62
//*************************************************************
//* SAMPLE CLI BIND JCL
//*************************************************
//JOBLIB   DD  DISP=SHR,DSN=DB2V61H1.SDSNEXIT
//         DD  DISP=SHR,DSN=DSN610.SDSNLOAD
//BINDCLI  EXEC PGM=IKJEFT01,DYNAMNBR=20
//DBRMLIB  DD DISP=SHR,DSN=DSN610.SDSNDBRM
//SYSTSPRT DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSTSIN  DD  *
 DSN SYSTEM(DBH1)
 RUN PROGRAM(DSNTIAD) PLAN(DSNTIA61) LIBRARY ('DB2V61H1.RUNLIB.LOAD')
 BIND PACKAGE (DSNAOCLI) MEMBER(DSNCLICS) ISOLATION(CS)
 BIND PACKAGE (DSNAOCLI) MEMBER(DSNCLINC) ISOLATION(NC)
 BIND PACKAGE (DSNAOCLI) MEMBER(DSNCLIRR) ISOLATION(RR)
 BIND PACKAGE (DSNAOCLI) MEMBER(DSNCLIRS) ISOLATION(RS)
 BIND PACKAGE (DSNAOCLI) MEMBER(DSNCLIUR) ISOLATION(UR)
 BIND PACKAGE (DSNAOCLI) MEMBER(DSNCLIC1)
 BIND PACKAGE (DSNAOCLI) MEMBER(DSNCLIC2)
 BIND PACKAGE (DSNAOCLI) MEMBER(DSNCLIF4)
 BIND PACKAGE (DSNAOCLI) MEMBER(DSNCLIMS)
 BIND PACKAGE (DSNAOCLI) MEMBER(DSNCLIQR)
 BIND PLAN(DSNACLI)                    -
   PKLIST(DSNAOCLI.DSNCLICS            -
          DSNAOCLI.DSNCLINC            -
          DSNAOCLI.DSNCLIRR            -
          DSNAOCLI.DSNCLIRS            -
          DSNAOCLI.DSNCLIUR            -
          DSNAOCLI.DSNCLIC1            -
          DSNAOCLI.DSNCLIC2            -
          DSNAOCLI.DSNCLIF4            -
          DSNAOCLI.DSNCLIMS            -
          DSNAOCLI.DSNCLIQR            )

 END
//SYSIN DD *
   GRANT EXECUTE ON PLAN DSNACLI TO PUBLIC ;
/*
```

Figure 41. Sample JCL to bind the CLI modules

4.3.8.3 Settings for the CLI initialization file

The DB2 OLAP Server installation program automatically creates the CLI initialization dsnaoini file for the use of DB2 OLAP Server in the /etc/essbase directory. These settings allow the Recoverable Resource Manager to make multiple concurrent connections using CLI. The MVSDEFAULTSSID parameter must be set with the DB2 subsystem ID.

```
[COMMON]
MULTICONTEXT=1
CONNECTTYPE=1
MVSDEFAULTSSID=DBH1
[DBH1]
MVSATTACHTYPE=RRSAF
```

Figure 42. Sample dsnaoini configuration file

The installation program also creates the RSM configuration file rsm.cfg in the /etc/essbase directory. This file contains the DB2 parameters you entered during the setup phase used by the DB2 OLAP Server. The RDB_NAME parameter must be set with the DB2 location name. For detailed information about the parameters in rsm.cfg, refer to *Using DB2 OLAP Server*, SC26-9235.

```
[RSM]
RDB_NAME=DBH1
TRACELEVEL=0
ISOLATION=CS
TABLESPACE=IN OLAPDB.OLAPTS
ADMINSPACE=IN OLAPDB.ADMINTS
```

Figure 43. Sample rsm.cfg configuration file

4.3.9 Perform the CLI test

DB2 for OS/390 Call Level Interface (CLI) works with DB2 OLAP Server. You need to be sure that CLI works correctly:

1. Check the dsnaoini file in the /etc/essbase directory. It has been customized by the installation process.
2. Ensure that the DB2 subsystem and RRS are started.
3. Execute the CLITEST test in the directory /u/essbase/rsm/bin.

You are prompted to enter your database name, which is the RDB_name in the rsm.cfg configuration file. See the sample dialog in Figure 44.

```
BI390B:/u/essbase/rsm/bin: >CLITEST
**** Entering CLITEST.

Please enter database name: DBH1

SQLAllocEnv was successful.
SQLAllocConnect was successful.
SQLConnect was successful.
SQLGetInfo was successful.
Multiple connection support is enabled.
Only the local DB2 can be used with the SQL Interface function.

**** CLITEST succeeded.

**** Exiting CLITEST.
BI390B:/u/essbase/rsm/bin: >
```

Figure 44. Sample execution of CLITEST program

You must get the message `Multiple connection support is enabled` for a successful connection. If not, check the CLI or the RRS configuration, or your dsnaoini configuration file.

Notes: If you get an error message such as `The module DSNAOCLI was not found`, you should execute the profile file(.profile) in the DB2 OLAP Server home directory (/u/essbase/rsm), then run CLITEST again.

4.3.10 Create a database and tablespaces for DB2 OLAP Server

You now need to create a DB2 database environment that will be used for DB2 OLAP Server to store DB2 OLAP server catalog information and the data for sample applications. For your own application, you can create databases and tablespaces later. Refer to 7.8.2.1, “Allocate a separate DB2 database for the application” on page 188.

You must have the authorization to create a DB2 database and grant authority to others, such as SYSADM or CREATEDBA.

- Create a database and the tablespaces. You need to specify a 32 KB administrative tablespace, ADMINTS.

- You must create segmented tablespaces. If not, you will get errors when saving the DB2 OLAP Server outlines.
- Use the values in Figure 45 on page 91. Do not allocate less PRIQTY and SECQTY than the values in the example.
- Specify the same database and tablespace names you defined in the rsm.cfg file (TABLESPACE, ADMINSPACE parameters); see Figure 43 on page 88.
- Grant the appropriate DB2 authority to the DB2 OLAP Server supervisor ID or RDB_USERID in rsm.cfg file.

The sample JCL in Figure 45 on page 91 creates the DB2 database and two tablespaces. DB2 OLAP Server stores all non-administrative table data in the OLAPTS tablespace, and all administrative tables in ADMINTS. For more detail information on the rsm.cfg configuration file, refer to *Using DB2 OLAP Server*, SC26-9235.

- Run this job, and you should get return code equal to 0.

Important

DB2 OLAP Server for OS/390 does not support DB2 partitioned tablespaces. You must create segmented tablespaces.

The fact table is physically split in four different fact tables (numbered from 1 to 4). To improve performance, you can use the rsm.cfg configuration file parameters to separate the allocation of these fact tables in different tablespaces. Refer to 7.8, “How to optimize data placement” on page 186.

You can also use the DB2 OLAP Server Partitioning Option to create separate physical cubes.

```
//BI390B  JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=BI390B
//*************************************************************
//* SAMPLE JCL TO ALLOCATE THE DB2 OLAP SERVER DATABASE
//* AND TABLESPACES
//*************************************************************
/*JOBPARM SYSAFF=SC62
//STEP1   EXEC PGM=IKJEFT01
//STEPLIB  DD DSN=DB2V61H1.SDSNEXIT,DISP=SHR
//         DD DSN=DSN610.SDSNLOAD,DISP=SHR
//SYSTSPRT  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSUDUMP  DD SYSOUT=*
//SYSTSIN   DD *
 DSN SYSTEM(DBH1)
 RUN PROGRAM(DSNTIAD) PLAN(DSNTIA61) -
     LIBRARY('DB2V61H1.RUNLIB.LOAD')
 END
//SYSIN DD *
  CREATE STOGROUP SGOLAP
         VOLUMES (TOTDBK) VCAT DB2V61H1 ;
  CREATE DATABASE OLAPDB ;
  CREATE TABLESPACE OLAPTS IN OLAPDB
   USING STOGROUP SGOLAP
         PRIQTY 150000
         SECQTY 20000
         ERASE NO
         SEGSIZE 4 LOCKSIZE PAGE LOCKMAX SYSTEM CLOSE NO ;
  CREATE TABLESPACE ADMINTS IN OLAPDB
   USING STOGROUP SGOLAP
         PRIQTY 150000
         SECQTY 20000
         ERASE NO
         SEGSIZE 4 LOCKSIZE PAGE LOCKMAX SYSTEM CLOSE NO
         BUFFERPOOL BP32K ;
/*
```

Figure 45. Sample JCL to create the DB2 OLAP database and tablespaces

The DB2 OLAP Server supervisor ID must have DBADM authority on the DB2 databases where DB2 OLAP Server data will reside, in order to perform the following tasks:

- Create, drop, and alter tables
- Create and drop views
- Create and drop index

- Select, update, and delete data.

From SPUFI or a batch job (DSNTEP2 or DSNTIAD), issue the following command:

```
grant DBADM on OLAPDB to OLAPR;
grant use of STOGROUP stogroup-name to OLAPR;
```

4.3.11 Verify RACF password authorization for DB2 OLAP Server

You need to verify first that the `p` extended attribute is set for the PWDTEST and ESSBASE modules, located in the DB2 OLAP Server bin directory (/u/essbase/rsm/bin). From the OMVS environment, issue the commands:

```
ls -lE ESSBASE
ls -lE PWDTEST
```

You should get the result shown in Figure 46.

```
BI390B:/u/essbase/rsm/bin: >ls -lE ESSBASE
-rwxr-xr-x  -ps  1 STC      SYS1     1994752 Feb  7 16:25 ESSBASE
BI390B:/u/essbase/rsm/bin: >ls -lE PWDTEST
-rwxr-xr-x  -ps  1 STC      SYS1       77824 Jan 24 15:57 PWDTEST
BI390B:/u/essbase/rsm/bin: >
```

Figure 46. List of extended attributes for RSM program

If the `p` attribute is not set correctly, you must use the `extattr` command to set it:

```
extattr +p ESSBASE
extattr +p PWDTEST
```

To modify the program controlled extended attributes, your TSO user ID must be defined with the read permission to the CLASS(FACILITY) BPX.FILEATTR.PROGCTL profile.

You now need to test whether the user ID you use to log in to the DB2 OLAP Server is defined to RACF.

Execute the profile file you created previously in the DB2 OLAP Server home directory (/u/essbase/rsm), by issuing:

```
. .profile
```

Run the PWDTEST program to test the DB2 OLAP Server security options. Enter the user ID and password you have previously created for DB2 OLAP Server. Refer to Figure 47 for PWDTEST sample execution.

You must get a successful message.

```
BI390B:/u/essbase/rsm/bin: >PWDTEST

This program prompts for a RACF user ID and password. It then checks
that OS/390 UNIX is configured to interact with RACF using
the user ID and password provided.

Enter user ID:
OLAPR

Enter password:
OLAP390

The test was successful. OS/390 UNIX is correctly configured
to interact with RACF and the user ID and password are specified to
RACF.

BI390B:/u/essbase/rsm/bin: >
```

Figure 47. Sample execution of PWDTEST program for RSM

4.3.12 Start DB2 OLAP Server for OS/390

You are now ready to start DB2 OLAP Server for OS/390.

There are three different ways to start the server:

- In a foreground mode: This mode is required for the first start. Use OS/390 UNIX from a TSO session or from a telnet session to start the DB2 OLAP Server.
- In a background mode: Use OS/390 UNIX from a TSO session or from a telnet session to start the DB2 OLAP Server. The session is prompted after the server has started.
- Using an MVS started task procedure: This is recommended to run the DB2 OLAP Server in an operational environment. To set up an MVS started task procedure, refer to 6.1, “Activating the DB2 OLAP Server” on page 143.

To start DB2 OLAP Server for OS/390 do the following:

1. Log on to OS/390 UNIX from a TSO session or a telnet session. Make sure you run the .profile file created during the installation phase and stored in the DB2 OLAP Server home directory (/u/essbase/rsm).
2. Change to the DB2 OLAP Server bin directory (/u/essbase/rsm/bin).
3. Start the DB2 OLAP Server in the foreground by entering:

   ```
   ESSBASE
   ```

4. You are prompted to enter:
   - Your company name
   - The DB2 OLAP Server supervisor ID and password
5. Confirm the information you entered.

   After this initialization step, you should receive the message:

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

   Figure 48 on page 95 shows a sample dialog when starting the server *for the first time* in the foreground mode.

```
BI390B:/u/essbase/rsm/bin: >ESSBASE

Code page in use: IBM1047
11 login system
Hyperion Essbase OLAP Server - 5.0.2 Patch2 (03/19/99)
Copyright 1991-1998 Hyperion Solutions Corporation.
US Patent Number 5,359,724
All Rights Reserved.
Serial number:  180000050032639A-00BF203D63A
Welcome to the Hyperion Essbase OLAP Server.

Enter your company name Ý¨: IBM
Enter your name Ý¨: OLAPR
Enter the system password Ý¨: OLAP390

You have entered the following details:
Company name:     IBM
Your name:        OLAPR
System password:  OLAP390

These will be used to create the initial system security information
including the system supervisor.
Are these details correct? (y/n): Y

Waiting for Client Requests...
```

Figure 48. Starting DB2 OLAP Server with RSM

6. You can now stop the DB2 OLAP Server and choose to start it in background mode.
   - To stop the DB2 OLAP Server, enter the command:

     ```
     EXIT
     ```

   - To start the DB2 OLAP Server in background, enter the command:

     ```
     ESSBASE password -b&
     ```

   - To stop the DB2 OLAP Server started in background mode, use the ESSCMD `SHUTDOWNSERVER` command (Refer to 6.2, “Deactivating the DB2 OLAP Server” on page 147).

Refer to 4.5, “Install the sample applications” on page 97 to set up sample applications. Then, go to Chapter 5, “Setting up clients” on page 103, to install the client workstation options, such as Application Manager and Spreadsheet Add-ins.

4.4 DB2 OLAP Server maintenance procedure

This section describes the procedure to apply maintenance to the DB2 OLAP Server for OS/390 product.

When you receive PTFs to install on the DB2 OLAP Server product, do the following tasks:

1. Complete the SMP/E installation. This creates a DB2OLAP.Z file in the OS/390 UNIX file system /usr/lpp/essbase/tar directory.
2. Log on to OS/390 with a user ID that has read and write access to the DB2 OLAP Server directories.
3. If any applications are running, stop them by entering the `STOP appname` command from the Essbase Console or the `UNLOADAPP appname` from the ESSCMD command line, or by using the Application Manager.
4. Stop the DB2 OLAP Server process by entering `EXIT` from the console or the `SHUTDOWNSERVER` command from the ESSCMD command line.
5. Remove or rename the *390* directory created by the original installation of DB2 OLAP Server if it still exists.
6. Unpack the tar file that contains the PTF by entering the following command from the /usr/lpp/essbase/tar directory:

   ```
   tar xpfo DB2OLAP.Z
   ```

7. Change to the new 390 directory created by unpacking the DB2OLAP.Z file.
8. Run the essmaint program from the 390 directory.
9. When prompted, supply the directory name where the DB2 OLAP Server is installed. This is the directory name entered during the original product installation.
10. When the installation is complete, restart the DB2 OLAP Server.

Note

The PTF installation will create an API directory (if one does not already exist) under the directory you installed the PTF into. For example, if you entered /u/essbase/mdsm during the PTF installation it will create /u/essbase/mdsm//api.

If, during the installation of the DB2 OLAP Server, you chose to install the API files to a separate directory, delete those files and that directory and use the files in the newly created API directory.

Figure 49 shows a sample DB2 OLAP Server for OS/390 maintenance dialog.

```
/usr/lpp/essbase/tar/390 >essmaint

PTF installation program for IBM DB2 OLAP Server 1.1 for OS/390
Licensed Material - Property of IBM
Copyright (c) 2000 IBM. All right reserved.
US Government Users Restricted Rights -
Use, duplication or disclosure restricted
by GSA ADP Schedule Contract with IBM Corp.

You must agree to the terms of the IBM
International Program License Agreement
(IPLA) before installing or using this Program.
The IPLA is included in your program package.
Accept the terms of the Restricted Rights Legend described
above and continue with the install? (yes/no): yes
Please enter the directory where DB2 OLAP Server is installed
(/u/essbase/): /u/essbase/mdsm
Multidimensional storage manager detected.

Installing DB2 OLAP Server PTF, please wait...
Registering serial number: 180000050032639A-01FF204163F
Essbase Registration Utility - 5.0.2 Patch2 (03/19/99)
Copyright(c) 1991-1999 Hyperion Solutions Corporation
U.S. Patent Number 5,359,724
All rights reserved
Writing serial number 180000050032639A-01FF204163F
Writing license number into ESSBASE
Writing license number into ESSSVR

The installation will now attempt to set the 'p' extended attribute
for some of the installed files. If this fails alter the 'p'
extended attribute for the files shown, using the extattr command
from a user ID with appropriate authority.
DB2 OLAP Server for S/390 PTF installation finished successfully.
/usr/lpp/essbase/tar/390 >
```

Figure 49. DB2 OLAP Server for OS/390 maintenance dialog

4.5 Install the sample applications

DB2 OLAP Server for OS/390 provides four sample applications that can be used for testing. To install these applications on the server:

1. DB2 OLAP Server engine, either MDSM or RSM, must be activated before creating the sample applications.
2. Log on to an OS/390 UNIX session through TSO or telnet.
3. If you log on from TSO, make sure that your virtual memory address space size is set to at least to 256 MB. It is needed to create the sample applications.
4. Make sure that DB2 OLAP Server supervisor has appropriate authority in DB2 (RSM only). Refer to 9.2, “Administering DB2 security” on page 209.
5. Run the SAMPLE program from the bin directory (/u/essbase/rsm/bin or /u/essbase/mdsm/bin depending on which storage manager you have activated). Refer to Figure 50 on page 99 for the sample application installation dialog.
6. You are prompted to provide:
   - The DB2 OLAP Server hostname: IP address or host name
   - The DB2 OLAP Server supervisor ID
   - The DB2 OLAP Server supervisor password

This program creates four applications in the DB2 OLAP Server /home/app directory, as shown in Table 6.

Table 6. Sample applications installed with DB2 OLAP Server for OS/390

DB2 OLAP Server sample application	Database name
Demo	Basic
Sample	Basic
	Interntl
	Xchgrate
Samppeast	East
Samppart	Company

```
BI390C:/u/essbase/mdsm/bin: >SAMPLE

This program will create the sample applications.

Enter the hostname:
9.12.2.26
Enter DB2 OLAP Server supervisor ID:
OLAPM
Enter DB2 OLAP Server supervisor password:
OLAP390
Login successful.
Creating application Demo.
Creating database Basic.
....
Finished restructure.
Creating application Sample.
Creating database Basic.
....
Finished restructure.
Creating database Interntl.
....
Finished restructure.
Creating database Xchgrate.
....
Finished restructure.
Creating application Sampeast.
Creating database East.
....
Finished restructure.
Creating application Samppart.
Creating database Company.
....
Finished restructure.
BI390C:/u/essbase/mdsm/bin: >
```

Figure 50. Sample applications installation

If you want to test your environment with these sample applications, you need to load the data into the databases from the DB2 OLAP Server Application Manager. Refer to 5.4, “Loading data into the sample applications” on page 130 or *Essbase Installation Notes*, GC26-9237, to load the data into the sample applications.

If you get errors during the installation of the sample applications, we strongly recommend that you do the following:

1. Stop the DB2 OLAP Server.
2. Remove the DB2 OLAP Server essbase.sec file located in the /home/bin directory.

> **Attention**
>
> By removing the file essbase.sec, you will lose *all information* previously created on applications and users.

3. Remove your app directories, except for the _tmpinst directory.
4. Drop the DB2 tablespaces created during the installation phase and create them again (for RSM only).
5. Set TRACELEVEL=7 in the rsm.cfg file (for RSM only).
6. Start the DB2 OLAP Server (`ESSBASE` command).
7. Enter your company name, supervisor user ID and password.
8. Run the SAMPLE program again.
9. Review the rsmtrace.log file in the DB2 OLAP Server home directory to get more information on the error (for RSM only).

Partitioning applications

The Samppart and Sampeast applications are designed to exploit application partitioning. However, they are not ready to perform this exploitation even after running SAMPLE program. You should follow the instructions in Chapter 15 “Building and Maintaining Partitions” of *DB2 OLAP Server DB Administrator’s Guide Volume 1,* SC26-9238, and the instructions in the program directory of DB2 OLAP Server for OS/390.

4.6 What’s next

After you start DB2 OLAP Server, you should install the Application Manager on a client workstation and verify that you can connect to the server.

For the *first* logon to the Application Manager, you must use the DB2 OLAP Server supervisor ID and password you defined during the installation process.

The next step is to install a spreadsheet on client workstations or other front-end tools you want to use to access DB2 OLAP Server applications.

At this point, you can also install the DB2 OLAP Server optional components such as the Web Gateway interface, Hyperion Integration Server, Hyperion Wired for OLAP, and DB2 OLAP Server Tools Bundle products. Refer to Chapter 5, “Setting up clients” on page 103 for these steps.

Chapter 5. Setting up clients

To use IBM DB2 OLAP Server services from a client, you need the appropriate software. This chapter shows you:

- How to install DB2 OLAP Server client software such as:
  - Essbase Application Manager
  - Essbase Add-in feature: Lotus 1-2-3, Microsoft Excel 2000
  - Hyperion Wired for OLAP
  - Web Gateway
- How to load the data for sample applications

Our test operating system for the installation of client software was Microsoft Windows NT 4.00.1381 Workstation with Microsoft Service Pack 5. The spreadsheets Lotus 1-2-3 Release 9.5 for Windows N9.5.9904.1503 and Microsoft Excel 2000 (9.0.2720) were already installed.

You can install Essbase client software:

- From CD-ROM
- From an installation disk set
- From a network drive

We will describe how to install Essbase client software from CD-ROM. For more information about how to install the Application Manager from other media, see *IBM DB2 OLAP Server Installation Notes,* GC26-9237*.*

Before you install Essbase client software, you should check the hardware and software requirements for the installation. Refer to 2.1, “Hardware and software requirements” on page 19.

5.1 Application Manager installation

Hyperion Application Manager supports a client/server architecture that can maintain servers, applications, and databases from a single console.

Hyperion Essbase Application Manager includes tools for:

- Building and modifying OLAP structures
- Managing shared or distributed data including defining database partitions, data replication, and loading data
- Managing metadata

- Administering users and user groups
- Creating analytical calculations
- Controlling server processing

The installation of Hyperion Essbase Application Manager and Add-in features can be done in one step. We decided to do a step-by-step installation in order to show you the details.

5.1.1 Installation steps

1. Run **setup.exe**.
2. The Software License Agreement is displayed. Click **Exit**, if you do not agree; the installation will terminate. Select **Accept** to agree.
3. In the Question window select **Yes** if you want to update an earlier version of Application Manager, select **No** if this is the first installation. It is assumed that this is the first installation.
4. In the Essbase Products window, select both **Application Manager** and **Runtime Client**. See Figure 51 on page 105.
5. If you select *Application Manager* only, it will end with an error message when called. Click **Next** to continue.
6. In the Language window mark your preferred language and click **Next**.
7. Select **Local** in the Client Setup and Destination Directory window and choose the destination directory. Click **Next** to continue.
8. Select **TCP/IP compliant** in the Choose Network Protocol window, then click **Next**.
9. Click **Next** to Install Online HTML Documentation**.**
10. Click **Yes** to let setup update your environment.
11. If you do not agree, click **Back** to correct your selections or click **Cancel** to end the installation. Otherwise, click **Next** in the Confirm Choices window to start the installation.
12. During the installation, you can see the installation progress and Release Information for every selected product.
13. Click **Yes** to read the Readme.txt file.

.

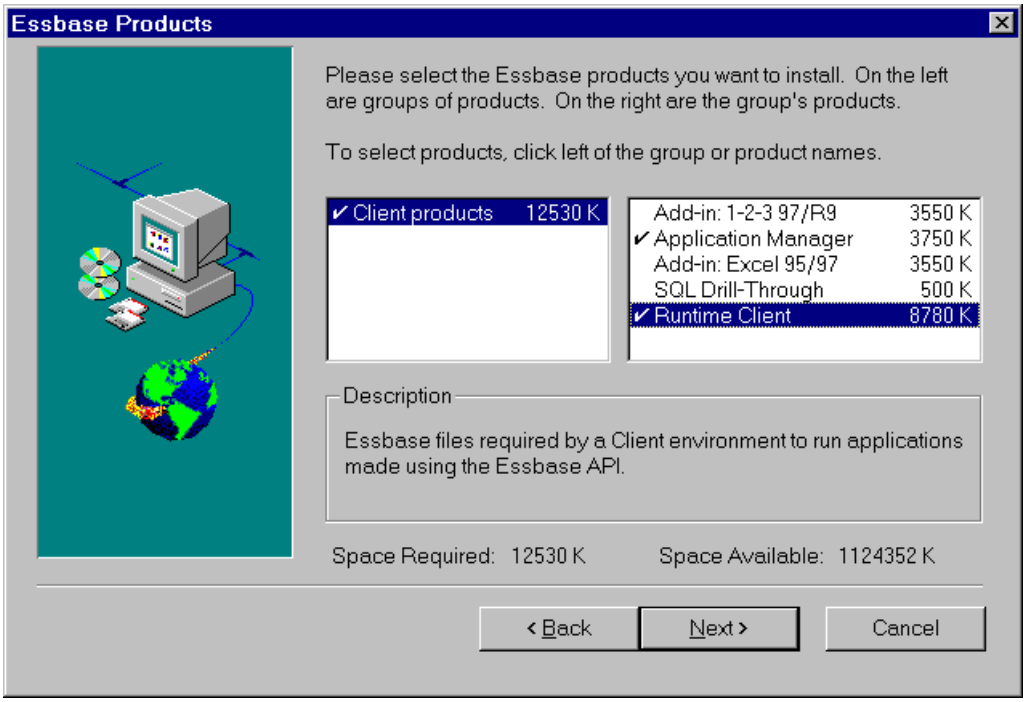


Figure 51. Essbase client products

14. The Information window indicates the successful completion of the installation.
15. Restart your system.

After restarting the system, you should find a new program group of IBM DB2 OLAP Server 1.1; see Figure 52.

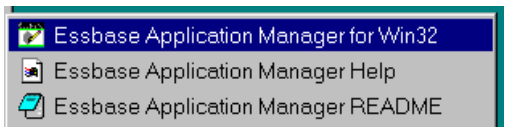


Figure 52. IBM DB2 OLAP Server items

You can activate the Application Manager through the following actions:

1. Be sure that the DB2 OLAP Server is active at the OS/390 system.

2. Click **Start--> Programs--> IBM DB2 OLAP Server 1.1.** You can see three entries as shown in Figure 52 on page 105. (If you install Add-in features at the same time, there will be one additional Add-In Help entry.)
3. Click **Essbase Application Manager for Win32** to start the Application Manager.
4. Click the item **Server** in the tool bar in the Application Manager window.
5. Select **Connect** out of the drop down list and enter:
   a. The Server Name or TCP/IP address of the DB2 OLAP Server.
   b. User name: at first connection to the Server, you must use the DB2 OLAP Server supervisor ID.
   c. Password: enter the password of the supervisor.
6. After you get a connection to your DB2 OLAP Server, it shows you a list of all applications on this server. At the first connection, you should see sample applications, such as Basic, Sample, Samppart, and Sampeast.

5.2 Add-in feature installation

In order to perform OLAP functions on the data in your business intelligence database, you need appropriate front-end software. You can use spreadsheets together with special Add-ins to display the results of these OLAP processes.

This section describes how to install and register the Add-ins provided with IBM DB2 OLAP Server for IBM Lotus 1-2-3 and for Microsoft Excel 2000.

5.2.1 Install and register Add-in for Lotus 1-2-3

There are two steps necessary to make the Add-in for Lotus 1-2-3 usable:

1. Install the software on your hard disk.
2. Register the Add-in for Lotus 1-2-3 to the main menu as a main item.

5.2.1.1 Install Lotus 1-2-3 Add-in on your hard disk

1. Run **setup.exe**.
2. The Software License Agreement is displayed. Select **Accept**, if you agree. Click **Exit**, if you do not agree; the installation will terminate.
3. In the Question window, select **No** if this is the first installation. Or select **Yes** if you want to update a previous version of Add-in for Lotus 1-2-3.

4. In the Essbase Products window, select **Add-in: 1-2-3 97/R9**, then click **Next**.
5. In the Language window, select your preferred language, then click **Next**.
6. In the Client Setup and Destination Directory window, select **Local.** If the path is ok, click **Next** to continue. If you want to enter another path, click **Browse...** and provide the drive and path, click **OK** to return, then **Next** to continue.
7. Select **TCP/IP compliant** in the Choose Network Protocol window, then click **Next**.
8. In the Confirm Choice window, read the information about the current settings. If you agree, click **Next** to continue. Otherwise, click **Back** to go to the previous window and correct your settings.
9. In the Readme window click **Yes** to read the Readme.txt, click **No** to continue.
10. The Information window tells you that the installation was successful. Click **OK** to finish the installation.

5.2.1.2 Register Add-in for Lotus 1-2-3:

1. Start Lotus 1-2-3.
2. Click on **File--> Add-Ins--> Manage Add-Ins** of the Lotus 1-2-3 main menu.
3. Click on **Register...**
4. Enter the path you installed the Lotus 1-2-3 Add-in at **Look in** and the name of the Add-in at **File name.**
5. Click **Open**.
6. Click the Essbase Add-in to mark it with a check mark as shown in Figure 53 on page 108.

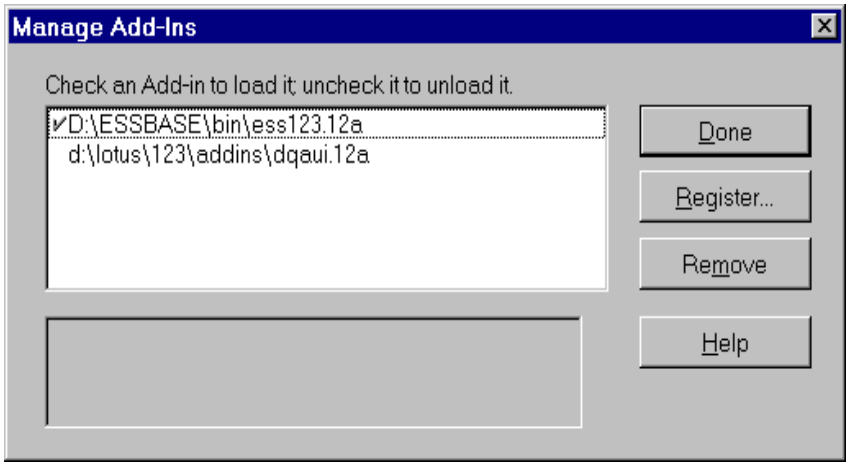


Figure 53. Lotus 1-2-3 Manage Add-Ins

7. Click **Done** to let Lotus 1-2-3 complete the registration. After successful registration, you will notice there is a new item, **Essbase,** in the tool bar of Lotus 1-2-3.

Figure 54 on page 109 shows the content of main menu item **Essbase** after the Add-in is loaded.

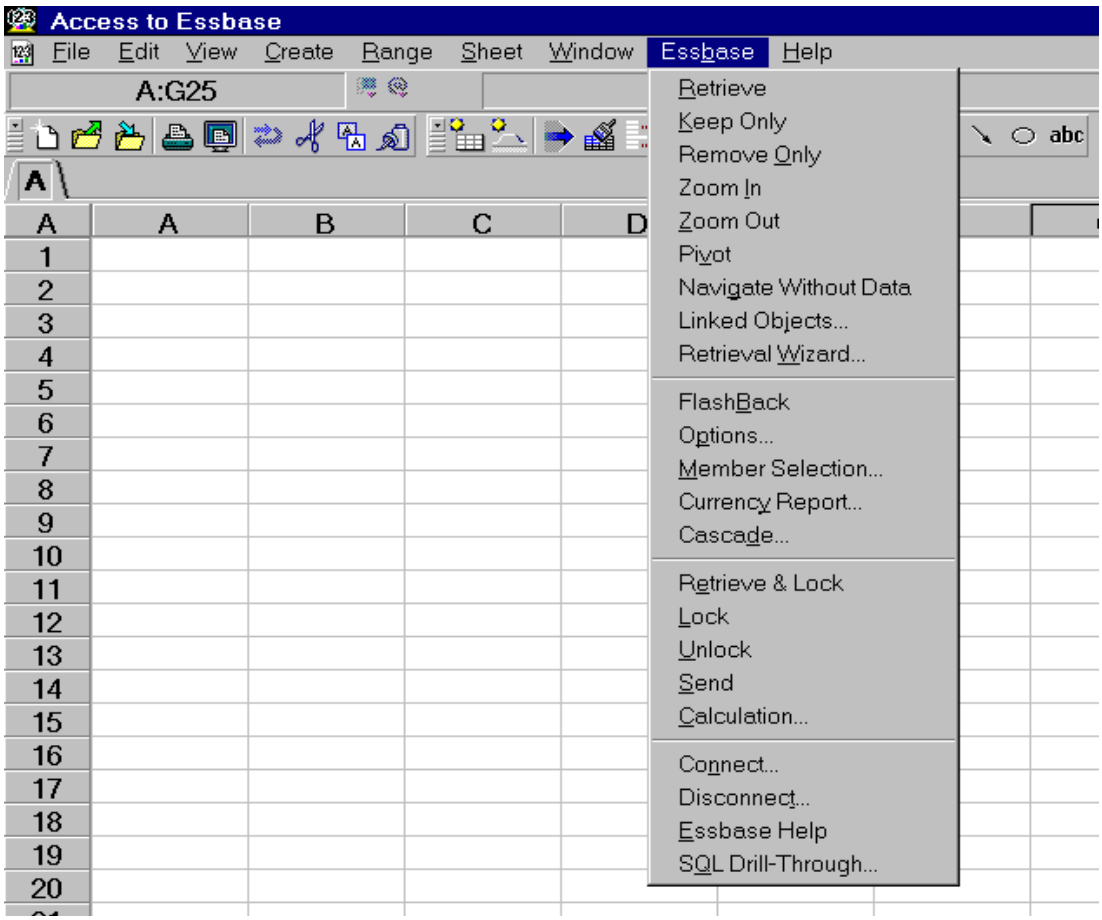


Figure 54. Lotus 1-2-3 Content of Essbase menu item

Note

The last item of the Essbase menu items shown in Figure 54, *SQL Drill-Through* is not present unless you install this product.

5.2.2 Install and register Add-in for Microsoft Excel 2000

There are two steps necessary to make the Add-in for Excel usable:

1. Install the software on your hard disk.

2. Register the Add-in for Micrsoft Excel 2000 to the main menu as a main item.

5.2.2.1 Install Add-in on your hard disk

1. Run **setup.exe**.
2. The Software License Agreement is displayed. Select **Yes**, if you agree. Click **No**, if you do not agree; the installation will terminate.
3. In the Question window, select **No** if this is the first installation. Or select **Yes** if you want to update an previous version of Add-in for MS EXCEL.
4. In the Essbase Products window, select **Add-in: Excel 95/97**, then click **Next**.

> **Note**
>
> Every time you start Excel, an Essbase Error window will inform you that the SQL Drill-Through custom Add-in could not be loaded. You can ignore this message.

5. In the Language window, select your preferred language, then click **Next**.
6. In the Client Setup and Destination Directory window, select **Local.** If the path is correct, click **Next** to continue. If you want to enter another path, click **Browse...** and provide the proper drive and the path. Click **OK** to return, then **Next** to continue.
7. Select **TCP/IP compliant** in the Choose Network Protocol window, then click **Next**.
8. In the Confirm Choice window, read the information about the current settings. If you agree, click **Next** to continue. Otherwise, click **Back** to go to the previous window and correct your settings.
9. In the Readme window, click **Yes** to read the Readme.txt, or click **No** to continue.
10. The Information window tells you that the installation was successful. Click **OK** to finish the installation.

5.2.2.2 Register Add-in for Microsoft Excel 2000

1. Start Excel.
2. Choose **Add-ins** from the item **Tools** in the main menu.
3. In the Browse window, enter the path you installed the Add-ins in at **Look in**.

4. The Add-in ESSEXCLN.XLL is listed in the Browse window; mark it and click **OK** to continue.
5. Select **Hyperion Essbase OLAP Server DLL** in the Add-Ins window as shown in Figure 55 and click **OK** to start the registration of the selected Add-in.

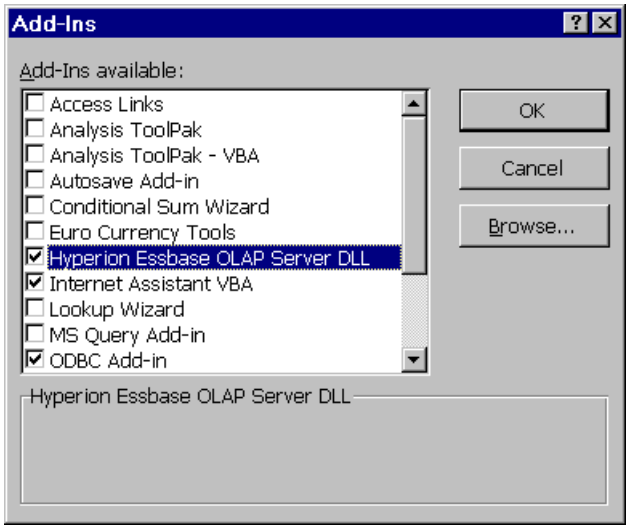


Figure 55. Microsoft Excel 2000 Add-Ins window

6. At successful registration, you will notice a new item, **Essbase,** is added to the Excel main menu. (For help on Add-in also see Microsoft Excel Online Help, keyword: add-in.)

Every time you start Excel, the Add-in is loaded and a new menu item **Essbase** is added to the main menu. Figure 56 on page 112 shows the content of this new item.

Note

The last item of the Essbase menu items shown in Figure 56 on page 112, *SQL Drill-Through,* is not present unless you install this product.

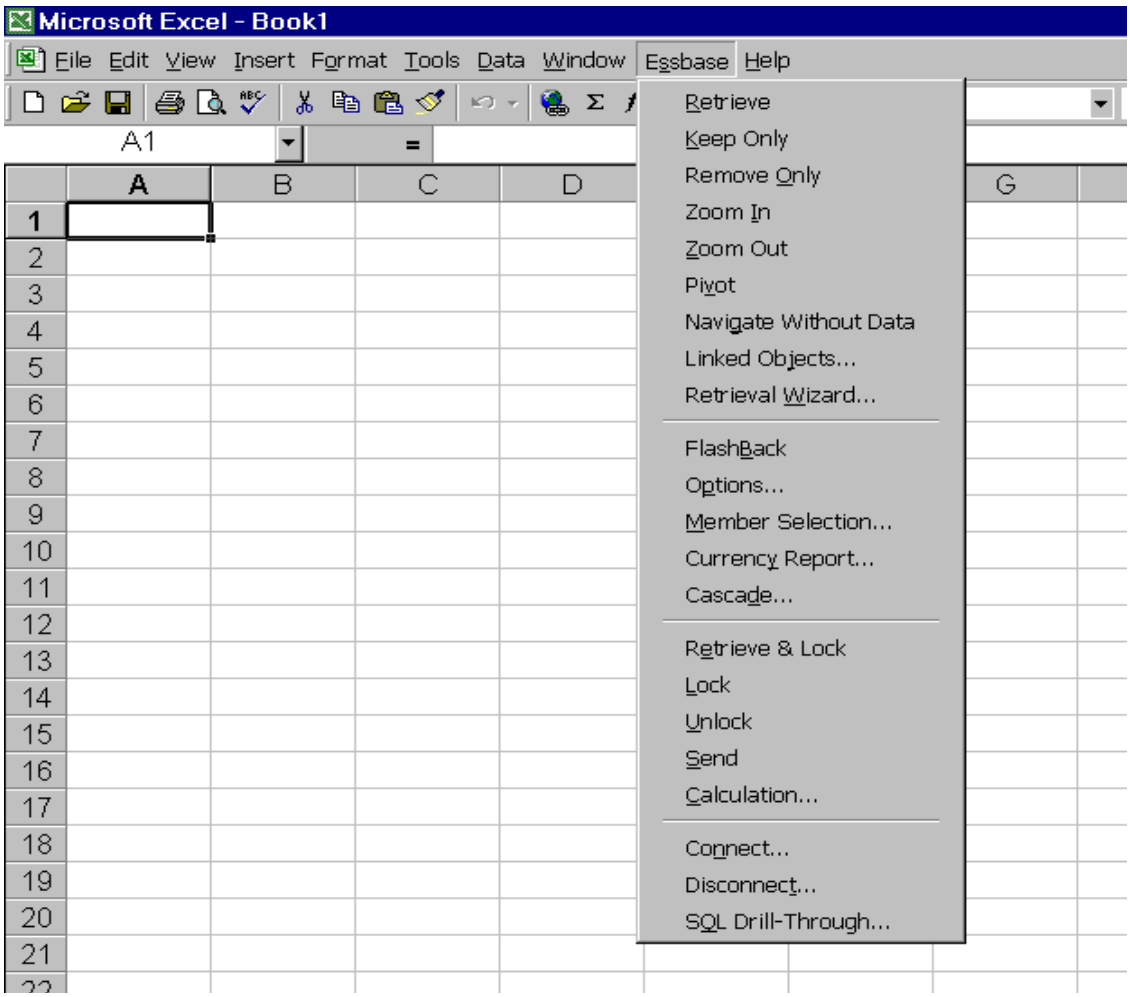


Figure 56. Microsoft Excel 2000 content of Essbase menu item

5.2.3 SQL Drill-Through

SQL Drill-Through, provided in DB2 OLAP Server for OS/390, is a sort of *server-based* SQL Drill-Through. All SQL Drill-Through requests for accessing a remote database are routed through the DB2 OLAP Server. When a user requests data from a relational database (this is DB2 for OS/390) using SQL Drill-Through, the request is sent from the Windows client to DB2 OLAP Server on OS/390. DB2 OLAP Server then routes the request through the DB2 ODBC interface to the DB2 databases on OS/390.

DB2 Connect is not required to use server-based SQL Drill-Through with DB2 OLAP Server on OS/390. Instead, you should install the ODBC feature of DB2 UDB for OS/390 and run a job that creates a plan that enables the ODBC interface. Refer to 4.3.8.2, “Enable the DB2 Call Level Interface (CLI)” on page 86 for more information.

For SQL Drill-Through that is *client-based*, then DB2 Connect is needed since the connection is initiated from an NT or UNIX machine to the DB2 database directly.

For more information about enabling SQL Drill-Through, refer to *Managing Multidimensional Data Marts with Visual Warehouse and DB2 OLAP Server*, SG24-5270, and *Hyperion SQL Drill-Through Guide,* SC26-9244.

5.2.3.1 How to install SQL Drill-Through

You can install SQL Drill-Through at the same time that you install other Add-in features (such as Lotus 1-2-3 Add-in, EXCEL Add-in). But you can also install it separately using the following steps:

1. Run **setup.exe**.
2. The Software License Agreement is displayed. Select **Yes**, if you agree. Click **No**, if you do not agree; the installation will terminate.
3. In the Question window, select **No** if this is the first installation. Or select **Yes** if you want to update a previous version of SQL Drill-Through.
4. In the Essbase products window, select **SQL Drill-Through,** then click **Next**.
5. In the Language window, select your preferred language, then click **Next**.
6. In the Client Setup and Destination Directory window, select **Local.** If the path is correct, click **Next** to continue. If you want to enter another path, click **Browse...** and provide the proper drive and the path. Click **OK** to return, then **Next** to continue.
7. Select **TCP/IP compliant** in the Choose Network Protocol window, click **Next**.
8. In the Confirm Choice window, read the information about the current settings. If you agree, click **Next** to continue. Otherwise, click **Back** to go to the previous window and correct your settings.
9. In the Readme window, click **Yes** to read the Readme.txt, click **No** to continue.
10. The Information window tells you that the installation was successful. Click **OK** to finish the installation.

5.2.3.2 Verifying the function of SQL Drill-Through

Once you install the SQL Drill-Through product, use the following steps to verify that all components are installed correctly:

1. Open either Lotus 1-2-3 or Excel for Windows. You use the Spreadsheet Add-in to access the SQL Drill-Through product.
2. If you are using Lotus 1-2-3, choose **File--> Open** and select the sheet SQLDRILL.WK4 located in the \Essbase\client\sample directory.

   If you are using Excel for Windows, open the sheet SQLDRILL.XLS in the same directory.

   These sheets are supplied as part of the Spreadsheet Add-in installation.
3. Choose **Essbase--> Connect** and connect to the Sample Basic database.
4. Select cell **C8** (value 232 or the data cell corresponding to cell C8), then choose **Essbase--> SQL Drill-Through**.

The SQL Database Login dialog box appears. The worksheet is made based on the dBASE files used, and you may not get any data by clicking the **Drill** button.

In order to verify SQL Drill-Through:

5. Click **Edit SQL** in the SQL Database login window.
6. Type an SQL request to retrieve data from any existing DB2 table, for example:

   ```
   Select * from DSN8610.EMP
   ```

   Then click **OK**.
7. Click **Drill** to begin an SQL Drill-Through retrieval. If successful, the drill action opens a new sheet containing the data you retrieve. You can limit the number of records to be read by clicking **Output option**.

5.3 Accessing OLAP server through the Web

In general, there two ways to access data in the OLAP database from Web clients:

- Using Hyperion Wired for OLAP
- Using Web Gateway

5.3.1 Installing and configuring Hyperion Wired for OLAP

Hyperion Wired for OLAP unites the enterprise through a scalable three-tier architecture:

- Upper-tier: IBM DB2 OLAP Server
- Mid-tier: Wired Repository
- Client-tier: Wired Analyzer as
  - Windows application: Standard or Professional Edition
  - Java Application: Web Viewer Version or Web Interactive Version

Wired for OLAP is built on a three-tier client/server architecture and gives access to OLAP data either from a Windows client or a Web client. Figure 57 shows the Wired for OLAP client/server architecture.

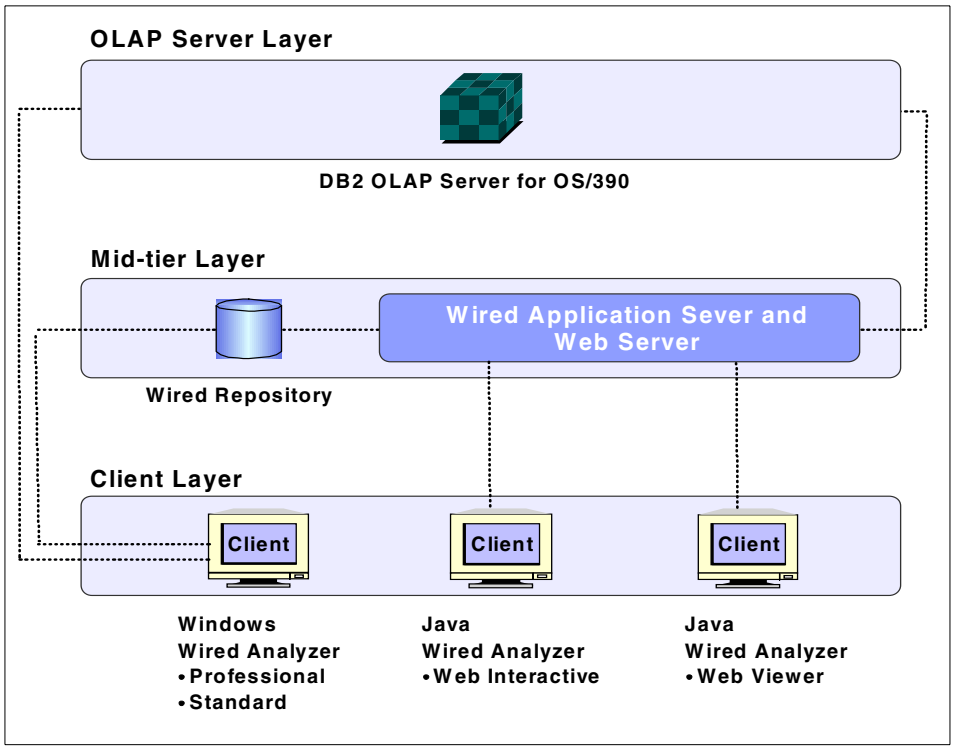


Figure 57. Wired for OLAP three-tier architecture

The client layer comes in different editions for Windows and the Web. The mid-tier Wired Application Server layer provides support for Windows and Web-clients. The Wired Repository stores and shares reports, user preferences, and other application-specific information.

Hyperion Wired for OLAP is a suite of applications. The complete suite consists of:

- Wired Analyzer
  - Web Viewer edition
  - Web Interactive edition
  - Standard edition
  - Professional edition
    - Wired Administrator
    - Wired Designer
    - Wired Excel Add-In
- Wired Application Server
- Wired Repository
- Enterprise Repository

5.3.1.1 Installation options

There are two Wired for OLAP installation options:

- **Evaluation Software** installs the Wired for OLAP Professional edition, Web Viewer, and Web View Interactive editions on a single machine for a 30-day evaluation period.
- **Licensed Software** requires an unlock code to install. The installation user must select one of two options:
  - The *end-user* option to install the Wired Analyzer - Standard Edition Windows client
  - The *administrator* option to install the entire Wired for OLAP system

Figure 58 on page 117 shows the available installation options.

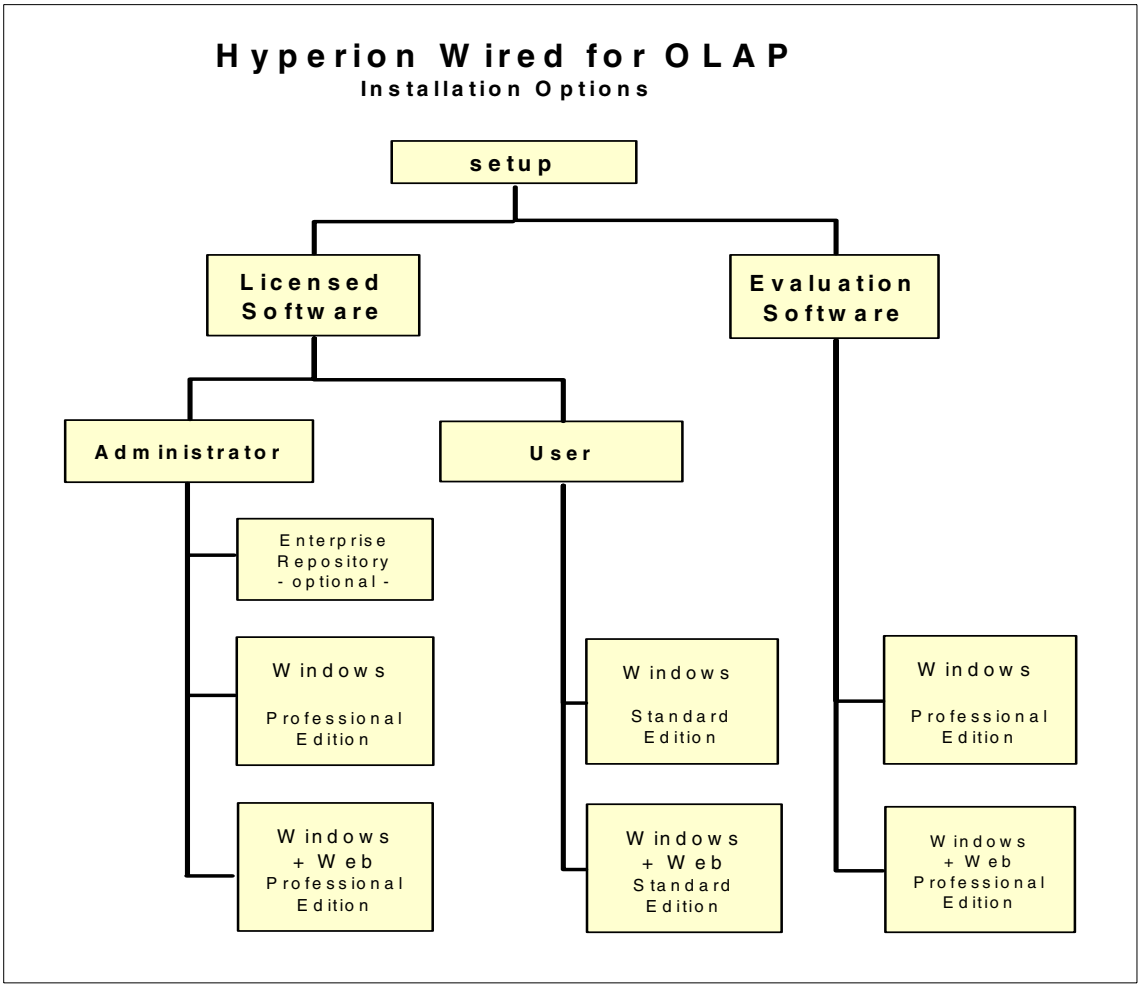


Figure 58. Hyperion Wired for OLAP - installation options

5.3.1.2 Evaluation Software - installation steps

The Wired for OLAP Installer displays two Evaluation Software Installation options:

- Windows: This installs the Wired Analyzer (professional edition), the Wired Application Server, and the Wired Repository for a 30-day evaluation period.
- Windows and Web: This installs the Wired Analyzer (professional edition), the Wired Application Server, the Wired Repository, and copies the Wired

Analyzer (Web Interactive edition or Web Viewer edition) for a 30-day evaluation period.

We do not describe the installation processes of the evaluation software. For more information about evaluation software, go to:

`http://www.hyperion.com`.

Note

If you install the evaluation software, keep in mind that the 30-day evaluation period means that on the *first* day of the next month you can no longer use Hyperion Wired for OLAP.

The license key registered to the evaluation software will license the software without uninstalling and running a complete setup again.

5.3.1.3 Licensed Software - installation steps

First of all, a license key is required to install Hyperion Wired for OLAP with the *Licensed Software option.*

Before installing the Licensed Software, you have to decide whether to use the administrator version or the user version (see Figure 58 on page 117).

The administrator version should be installed for advanced users or power users and Wired for OLAP administrators. You need at least one administrator version in your enterprise, because licensed users are dependent on licensed administrators for system access, user privileges, and repository location.

The user version should be installed for normal users. A user version is dependent on the repositories where the user is known. Whenever a user version of Wired for OLAP is started, you have to establish a connection to a repository. The computer the repository is installed on, the OLAP Server, and the Wired Application Server have to be running. Consider whether it makes sense to have the repository installed on a normal workstation.

The Wired Repository should be installed for enterprises with few users and only a small database. For large scale databases and many users, it is highly recommended to install the Enterprise Repository. This will reduce your administration efforts. In our test environment, we did not install the Enterprise Repository. A subsequent installation of the Enterprise Repository is possible and requires running the setup completely again. During this process, Hyperion Wired for OLAP will not delete existing components.

For detailed information on Wired for OLAP see *Hyperion for OLAP Getting Started Guide* and *Hyperion Wired for OLAP Enterprise Guide,* that are provided with the code.

To perform the installation, follow these steps.

1. Run the **setup.exe** of the Hyperion Wired for OLAP CD.
2. In the Wired for OLAP Installer window, click **Licensed Software.** For the Licensed Software installation, an unlock code (key/serial number) is required. See Figure 59.

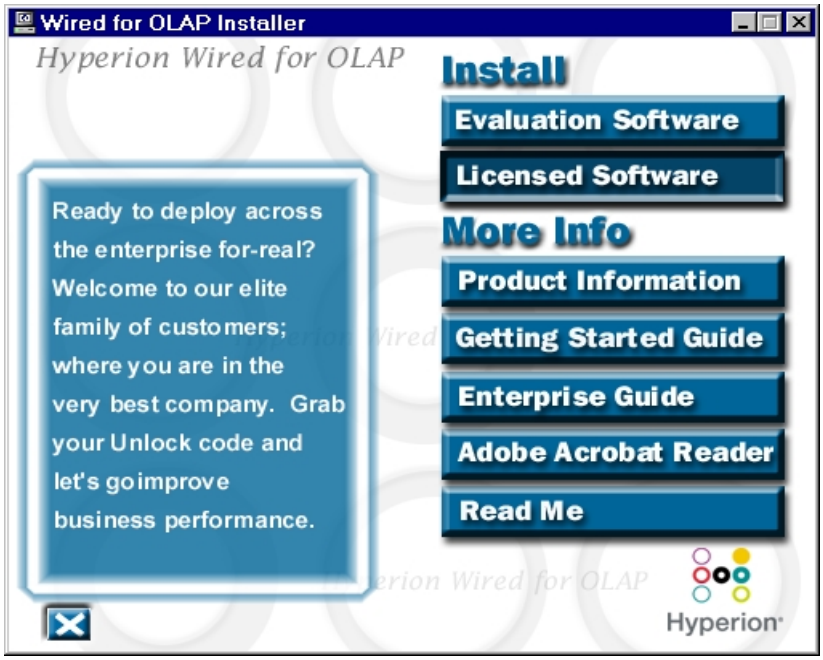


Figure 59. Wired for OLAP Installer start

3. Click **Administrator** to have the professional edition installed.
4. Click **Windows and Web** to start the installation.
5. In the Welcome window click **Next.**
6. In the User Information window, enter the serial number you received from Hyperion and click **Next**.

7. Click **Yes** in the Software License Agreement window to accept the License Agreement. Click **No** to terminate the installation; click **Back** to go to the previous window.
8. In the Choose Destination Location window click **Next** to accept the installation path. To change the installation path, click **Browse**. The Choose Folder window comes up. Select the drive, choose a directory or enter a new path, then click **OK** to return to the previous window. Click **OK** to continue.
9. If MS EXCEL is detected on your computer, a Question window comes up. You are asked to have a new menu item added to Excel. In case you do not have the correct version of Microsoft Virtual Machine, the installation returns to the first screen. You can add this functionality later. Click **No** to continue.
10. Read the text about repository and click **Next** to continue.
11. Choose **Wired Repository** in the Select Repository Type window, then click **Next**.
12. In the Information window, you are informed about the next installation step, read the text, then click **Next** to continue.
13. In the Repository Configuration window mark **Create New Repository** and click **Next**.
14. Click **Next** in the Select Program Folder window to accept the path name. This path name will be the *Wireddirectory*.
15. Click **Next** in the Start Copying Files window to copy the files to the destination folder. The installation progress is shown.

Attention

During our installation the set up informed us in the Self-Registration Failure Information window that *the self-registration of WiredCom.dll failed*, click **OK** to continue.

16. Click **No** in the Question window: do not import data from an earlier version.
17. Click **No** in the Question window: connections to available data sources can be created later. Click **Yes** to create the connection to an existing data source. In this case you need login information for each database.

18. The Setup Complete window tells you that installation is done. Select the option to read the Readme.txt, deselect the option to **Start Wired Analyzer**. Click **Finish**.

19. Setup is complete, click to **X** and confirm to exit.

After a successful installation of Hyperion Web Gateway, restart your system. In **Start--> Programs,** a new group **Hyperion Wired for OLAP** has been added.

5.3.1.4 Getting started with Wired Analyzer

You can start the Wired Analyzer in the following ways:

- From Windows
- From a Web browser using:
  - Web Viewer edition: This is the entry-level Wired for OLAP client. It is designed specifically for users that are focused on report-consumption. This streamlines the experience for these users by freeing them from a clutter of advanced features that would serve only to hinder, rather than help, them. A highly simplified interface enables users to quickly and easily access reports and navigate between them. Users can also change display types, print formatted reports, and transfer data to other applications. Where permitted by the report creator, users can also pivot dimensions, drill down for more detail, and drill-link to jump seamlessly to other reports.
  - Web Interactive edition: This is designed to address the needs of users who want the ability to interact more fully with reports, as well as easily consume and navigate them. Additional features available in the Web interactive edition include the ability to browse dimensions to make new selections, write-back edited data to OLAP servers, create new reports from existing ones, and save these new reports into their own personal report groups.

Note

You may only use products that you are licensed for. If you are not sure which product you should use, check with the person responsible for Wired for OLAP within your organization.

Before you start Wired Analyzer, be sure that the OLAP server is running and the application Drinks **(**or other applications) and its databases are loaded

and calculated. Refer to 5.4.4, “Preparing the Drinks application of Wired for OLAP” on page 136.

Getting started with Wired Analyzer from a window

1. Click **Start--> Programs--> Hyperion Wired for OLAP--> Wired Analyzer**.
2. Enter your user ID and password (defined in RACF), and click **OK**.
3. Select the Drinks application and the database Demo from the existing connections in the Databases window. (See 5.4.4, “Preparing the Drinks application of Wired for OLAP” on page 136.)
4. You can navigate Wired for Analyzer as follows:
   a. Drag and drop the dimensions into X-axis and Y-axis, and choose the fields you want to put in your reports.
   b. After you get a report, make a chart by clicking **DISPLAY--> CHART**.

Getting started with Wired Analyzer from a Web browser

1. Click **Start--> Programs--> Hyperion Wired for OLAP--> Wired Application Server**.
2. Run your Web browser.
   a. If you use Netscape, click **File--> Open Page**, and then click **Choose File** and choose Wireddirectory\www\index.html of the path you installed Wired for OLAP in.
   b. If you use Microsoft Internet Explorer, click on **File--> Open**, and then click on **Choose File** and choose *‘Wireddirectory’*\www\index.html of the path you installed Wired for OLAP in.
3. Click **Open** to have the Web page displayed in the Web browser. This enables you launch either of the two Web client versions of Wired Analyzer.
4. Click **Web Interactive edition** to build reports using Wired Analyzer.
   a. Enter your Wired user ID and password, which does not have to be defined in RACF, but should be defined by the Wired administrator. The first time you do this, use {system} user’s ID and password.
   b. Click **New** and choose the application and the database name in the existing connections.
   c. Enter the user ID and password (RACF-defined) for the database access.
   d. Navigate the reports and charts, then click **File** and save them.

5. To invoke the Web Viewer edition from item 3:
    a. Enter the Wired user ID and password, which does not have to be defined in RACF, but should be defined by the Wired administrator. The first time you do this, use {system} user's ID and password.
    b. Click **Home** and select one of the Wired report groups. These report groups are made by the Wired Interactive edition at item 4-d.
    c. Enter the user ID and password (RACF-defined) for the database access.
    d. Choose the report in the selected group, and you can review the content or print it.

5.3.2 Installing the Web Gateway

The Web Gateway is a Web deployment platform to build Web applications. It is an interactive interface for viewing multidimensional data from a Web browser.

Using one of the common Web browsers, Web Gateway enables you to apply all the OLAP capabilities through the WWW. This includes access to different OLAP Servers via Internet, intranet, or extranet.

This client product makes Essbase multidimensional data instantly available to a wide number of users and sites, both local and remote, whose security privileges you control.

The Web Gateway is compatible with standard World Wide Web editing and administration tools. It uses templates for data retrieval and manipulation. You create these templates with any text editor; an HTML authoring tool is very helpful, but not mandatory. Web Gateway provides examples of templates and Web Gateway HTML tags to help you get started.

Figure 60 on page 124 shows how Web Gateway works:

1. A user starts up Web Gateway.
2. The login procedure starts with a Login page (that is a Web Gateway HTML template).
3. The Login page starts the esscgi program.
4. Web Gateway communicates requested login information to the Essbase server. The Essbase server checks the requested user name, password, and server against its own access information (including user privileges). It also checks application and database information.

5. The Essbase server communicates via Web Gateway to esscgi that the server successfully completed the connection, or returns the appropriate error messages.
6. The user requests Essbase data through an Web Gateway template.
7. EssCGI processes the request. Web Gateway then takes the request to the Essbase server.
8. The Essbase server returns the data in a table to Web Gateway.
9. Web Gateway converts the data into HTML format and passes it to esscgi.
10. Esscgi inserts the returned data into the appropriate template, and sends the information back to the Web browser for display to the user.

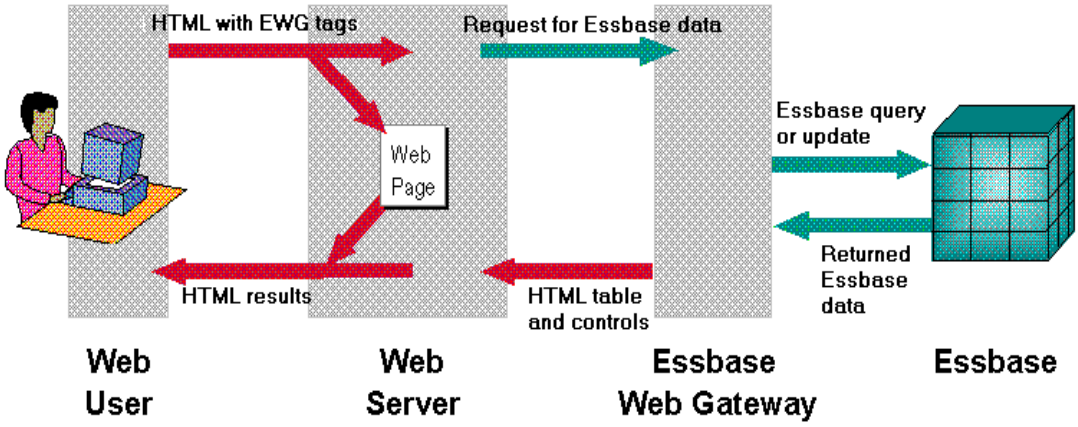


Figure 60. Simplified summary of how Essbase Web Gateway works

5.3.2.1 Installation steps for Web Gateway

This section describes how to install the Web Gateway software. There are two steps to make Web Gateway usable:

1. Installing the code
2. Registering Web Gateway

Note

We recommend that you do *not* install the Web Gateway and Essbase OLAP Server in the same directory. However, if for some reason both are installed on the same machine, the path for the OLAP Server has to be provided first and then the path for the Web Gateway.

Installing Web Gateway

1. Run **setup.exe** from NT.X86\DISK1 of the CD-ROM to start the installation.
2. In the Accept terms window click **Yes** to accept and continue, click **No** to terminate the installation.
3. In the Install where? window:
   a. To accept the suggested Destination Directory, click **Next.**
   b. If you do not want the suggested directory, click **Browse.** The Choose Directory window appears. Enter the installation path of your choice, then click **OK**. If the installation directory does not exist, the Setup window appears. Click **Yes** to have it created, click **No** to return to the Choose Directory window to enter another installation path. Click **Next** to start the installation.
4. If a previous version is detected, click **Yes** in the Overwrite? window to have it overwritten, or **No** to return to Install where?.
5. In the Confirmation window, click **Yes** to continue or **No** to return to the Install Where? window.
6. After all files are copied to their destination window, click **Yes** in the Update Environment window to have your environment updated by the installation program, or click **No** to do it manually (see “Manually updating the environment settings on Windows NT” on page 126).
7. In the Readme window, click **Yes** to read the Readme.txt, or click **No** to refuse.
8. The Information window tells you that the installation completed successfully.

Note

The installation does not create any icon in your desktop or any group in **Start** or **Start--> Programs.**

Registering Web Gateway

This section provides information on the Essbase server software licensing scheme for Web Gateway.

In addition to installing the Web Gateway software, and configuring your Web server, you must also enable Web Gateway on the DB2 OLAP Server. When you receive your Web Gateway software, you may be given a new DB2 OLAP Server software license number. Register your license number before you install Web Gateway.

To register your license number:

1. Stop the DB2 OLAP Server.
2. Run the Registration utility, REGISTER.EXE. You can find this in the ARBORPATH\BIN directory on the server, where ARBORPATH is your DB2 OLAP Server home directory. Enter the license number exactly as it is written. Do not use spaces.
3. Start the DB2 OLAP Server.

After you have installed Essbase Web Gateway, complete the following steps:

1. If you want to use the Web Gateway HTML documentation and examples, you must move or copy the EXAMPLES and DOCS directories and their contents to a directory directly below your Web server's document root.
2. Configure your Web server to run CGI Programs, and copy esscgi.exe into the appropriate directory or directories.
3. Create directories for your Web Gateway applications (HTML templates). These must be subdirectories of your Web server's document root.
4. If required, set the automatic logout time when you start Essbase Web

Manually updating the environment settings on Windows NT

The installation program gives you the option to update your environment automatically. To update the environment, the installation program changes one item in the Windows NT Registry.

If you do not want the installation program to update the Windows NT Registry, you can update it manually using the following steps:

1. Start Windows NT.
2. In **Start--> Settings** open the **Control Panel**.
3. Double-click the **System** icon. The System Properties window appears.
4. Click **Environment** and select the **Path** variable.
5. In the Value text box at the bottom of the System variables box, add the following to the end of the path variable:

   `C:\EWG\BIN`
6. If you installed Essbase Web Gateway to a different directory from the default, type that directory instead of C:\EWG\BIN.
7. Add the variable ARBORPATH. Type `ARBORPATH` in the Variable text box and the correct path in the Value text box (for example, c:\ESSBASE if you installed Web Gateway and the Essbase server on the same machine).

   Note: If you are using a dual-boot Windows NT and Windows 95 machine to run Essbase Web Gateway and Essbase, check that the ARBORPATH variable is correctly set in your AUTOEXEC.BAT file.
8. Click the **Set** button to update the path.
9. Click the **OK** button to close the System properties box.

Getting started

Before you begin, make sure you completed the tasks listed in Web Gateway Installation steps. It is important that your templates correctly reference the esscgi.exe file.

1. Make sure the IBM DB2 OLAP Server, the Web server, and Web Gateway are running.
2. You may want to design and implement a way for users to access Web Gateway by creating a home page for your application that has a link to an Web Gateway Login template.
3. Create a Login template. The Login template runs esscgi and starts the connection to Essbase when the user clicks the Login button. You can design your Web Gateway home page to link to the Login page, or simply have the Login page as the first page users access. Alternatively, a user could specify the entire path to the Login page. For example, a user might go directly to a login template by entering:

   `http://machine_name/ewg/login.htm`

   where `machine_name` is the Web server machine name.

4. Store the Login template in a directory directly below document root level on your Web server. You may want to name the directory templates and store all your Web Gateway templates there.
5. Create other templates according to your design. For example, from the Login template, Web Gateway automatically takes the user to a Select or Retrieve template, depending on how you set up the Login template. Store the templates in a directory directly below document root level on your Web server (see Step 4).

Each template can call other templates. For example, the Retrieve and Update pages can display a Query button. When the user clicks the Query button, Web Gateway displays a Query template, which displays a list of reports to choose from.

To call or invoke one template from within another, include the Web Gateway template tag for the called template in the calling template. For example, suppose you invoke the Retrieve template from the Login template. In the Login template, you would include the tag:

```
<INPUT NAME="EssRetrieveTemplate" TYPE=hidden

   VALUE="../template_directory/filename.htm">>
```

Where template_directory is the name of your template directory, and filename is the name of your Retrieve template. (This example references the called template file, filename, relative to the location of esscgi.exe. Your configuration may differ from this example.)

6. Test your templates to make sure the file references and other details are correct.

Running Web Gateway on Windows NT

The Web Gateway supports Windows NT registered service, so that Web Gateway starts automatically when you start Windows NT. You can choose whether to run Web Gateway as a Windows NT registered service.

To register and unregister your Web Gateway server as an Windows NT registered service, ensure that you have the executables such as ewgreg.exe, ewgunreg.exe, and ewg.exe in your EWGINST directory, where EWGINST is the directory into which you installed Essbase Web Gateway. If you installed Web Gateway into the default directory, this is C:\EWG\BIN.

Note

ewg.exe is the Windows NT registered service equivalent of essweb.exe.

To register Web Gateway, type the following on the command line:

```
ewgreg ewginst\ewg.exe
```

Where `ewginst` is the directory into which you installed Essbase Web Gateway. If you installed Web Gateway into the default directory, this is C:\EWG\BIN.

To start Web Gateway as a service:

1. If you are using Windows NT 4.0, choose **Settings--> Control Panel--> Services** from the Start menu. (If you are using an earlier version of Windows NT, choose **Main--> Control Panel--> Services**.)
2. Click **Arbor Software Essbase Web Gateway 2.0.**
3. Click the **Start** button.

To unregister Web Gateway, type the following on the command line:

```
ewgunreg
```

To run Web Gateway from the command line, and not as an Windows NT registered service:

1. Start your Web server. (For more information, see your Web server documentation.)
2. At a command prompt, go to the EWGINST\BIN directory, where EWGINST is the directory where you installed Essbase Web Gateway. If you installed Web Gateway into the default directory, this is C:\EWG\BIN. At the command prompt, type:

   ```
   essweb.exe
   ```

Note

If required, you can set an automatic logout time.

5.4 Loading data into the sample applications

To load data into OLAP databases, you can use Application Manager or ESSCMD. This section describes loading data into the sample applications that are delivered with IBM DB2 OLAP Server and Hyperion Wired for OLAP using Application Manager.

5.4.1 Preparing the Sample application

The IBM DB2 OLAP Server includes an application named Sample. In this section, we describe the loading of data to the databases of the Sample application.

5.4.1.1 Preparing the Basic database

The first database in the Sample application is called Basic. To prepare it for use, you need to load data into it. Complete the following steps:

1. Start the Application Manager.
2. Connect to the server in which the sample applications are installed.
3. Select the application called **Sample** from the Applications list box.
4. Select the database called **Basic** from the Databases list box; see Figure 61 on page 130.

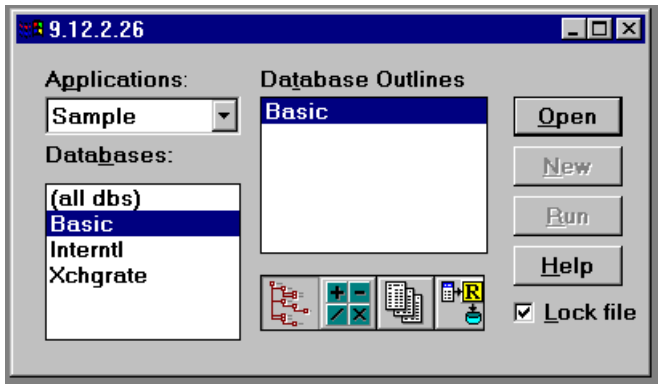


Figure 61. Application and Database list box

5. Choose **Database--> Load Data**. The Data Load dialog box appears as shown in Figure 62 on page 131:

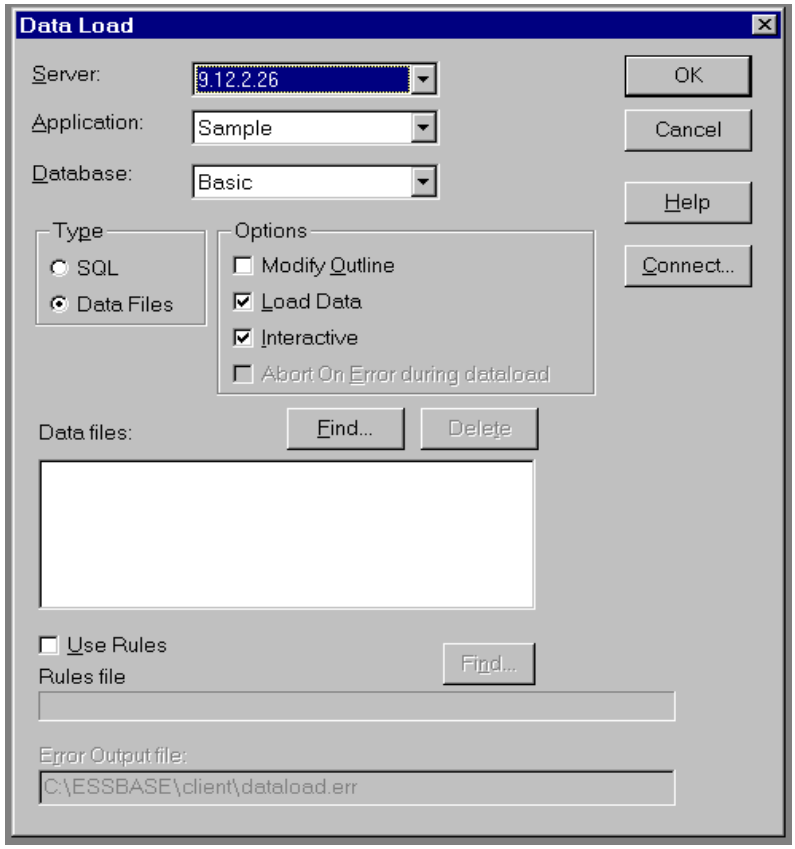


Figure 62. Data Load dialog box

6. Click the **Find** button shown in Figure 62 to locate a data file to load. The Open Server Data File Objects dialog box appears as shown in Figure 63 on page 132.

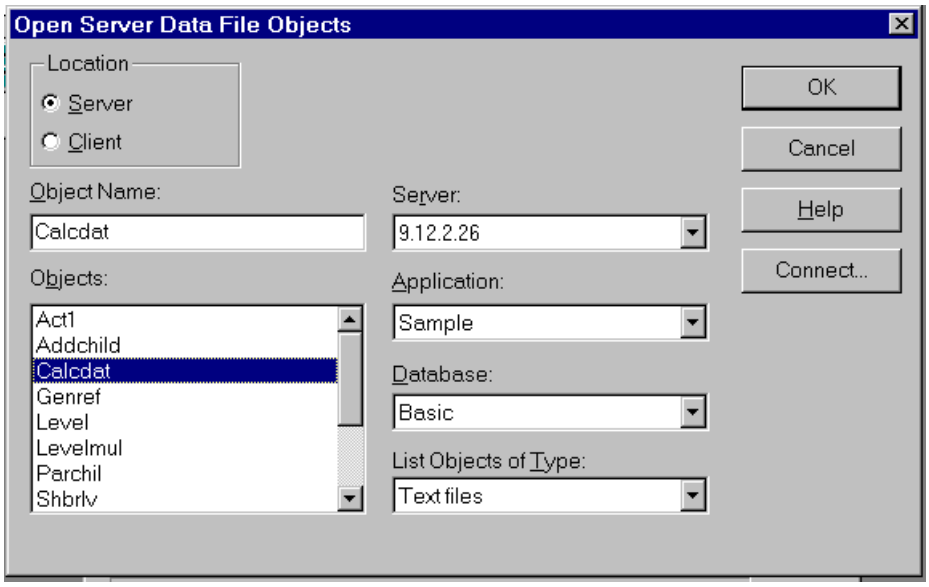


Figure 63. Selecting the CALCDAT text file

7. Ensure that List Objects of Type is Text files.
8. Select the text file named **Calcdat** from the Objects list box.
9. Click **OK** to close the dialog box. Essbase updates the Data Load dialog box to display the data file you chose; see Figure 64 on page 133.

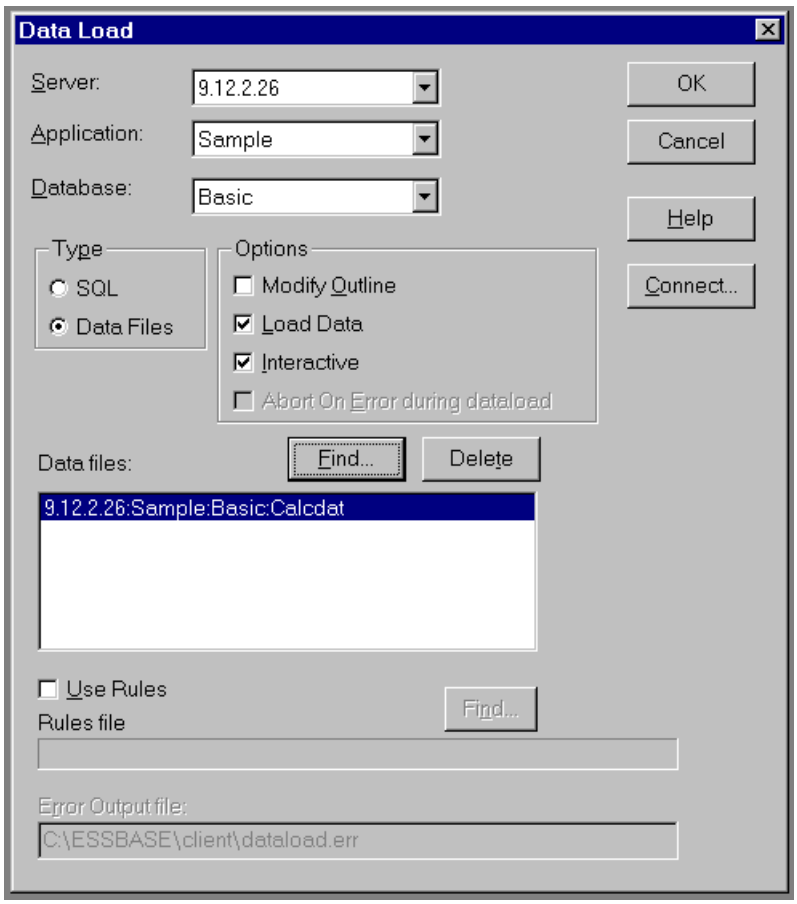


Figure 64. Data Load box with CLACDAT data file to select

10. Click **OK** to load the selected file. The data file is a large, fully calculated data set. It should take several minutes to load, and then the Data Load Completed dialog box appears. Click **Close**.

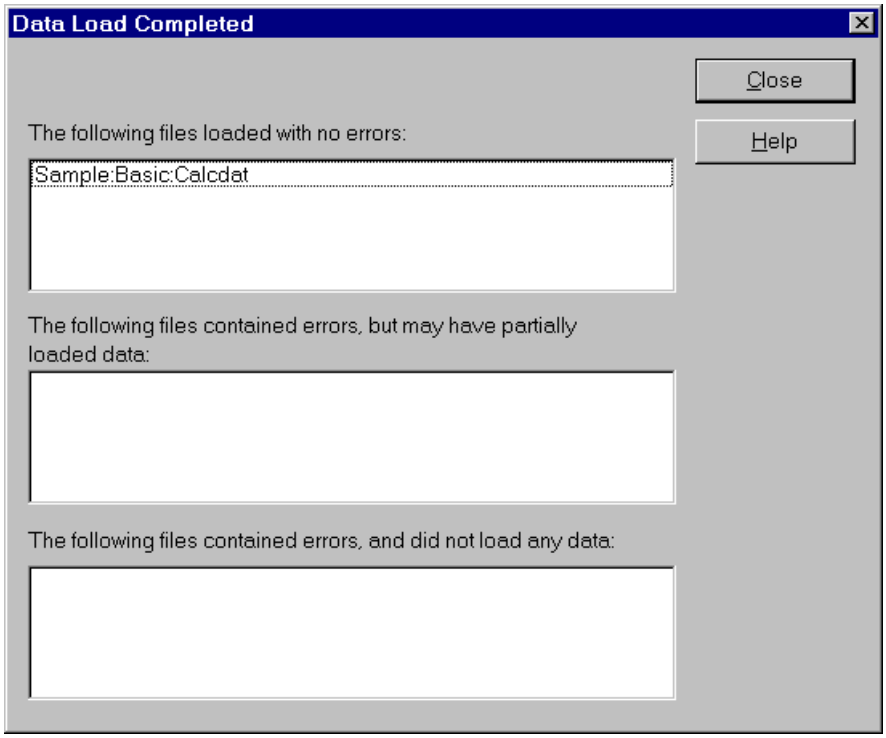


Figure 65. Data Load Completed information box

5.4.1.2 Preparing the INTERNTL database

The second database in the Sample application is called Interntl. To prepare it for use, you need to load data into it. Complete the following steps:

1. Start the Essbase Application Manager.
2. Connect to the server in which the Sample application is installed.
3. Select the application called **Sample** from the Applications list box.
4. Select the database called **Interntl** from the Databases list box.
5. Choose **Database--> Load Data**. The Data Load dialog box appears.
6. Click the **Find** button above the Data files list box to locate a data file to load. The Open Server Data File Objects dialog box appears.
7. Ensure that List Objects of Type is Text files.
8. Select the text file named **Currcalc** from the Objects list box.

9. Click **OK** to close the dialog box. Essbase updates the Data Load dialog box to display the data file you chose.
10. Click **OK** to load the selected file. The data file is a large fully calculated data set. It should take several minutes to load, and then the Data Load Completed dialog box appears. Click **Close**.

5.4.1.3 Preparing the Xchgrate Database

The third database in the Sample application is called Xchgrate. To prepare it for use, you need to load data into it. Complete the following steps:

1. Start the Application Manager.
2. Connect to the server in which the Sample application is installed.
3. Select the application called **Sample** from the Applications list box.
4. Select the database called **Xchgrate** from the Databases list box.
5. Choose **Database--> Load Data**. The Data Load dialog box appears.
6. Click the **Find** button above the Data files list box to locate a data file to load. The Open Server Data File Objects dialog box appears.
7. Ensure that List Objects of Type is Text files.
8. Select the text file named **Rates** from the Objects list box.
9. Click **OK** to close the dialog box. Essbase updates the Data Load dialog box to display the data file you chose.
10. Click **OK** to load the selected file. The data file contains currency exchange rates, and should take less than one minute to load. The Data Load Completed dialog box appears. Click **Close**.

5.4.2 Preparing the Demo application

The Demo application contains one database called Basic. To prepare it for use, you need to load data into it. Complete the following steps:

1. Start the Application Manager.
2. Connect to the server in which the sample application is installed.
3. Select the application called **Demo** from the Applications list box.
4. Select the database called **Basic** from the Databases list box.
5. Choose **Database--> Load Data**. The Data Load dialog box appears.
6. Click the **Find** button above the Data files list box to locate a data file to load. The Open Server Data File Objects dialog box appears.
7. Ensure that List Objects of Type is Text files.

8. Select the text file named **Data** from the Objects list box.
9. Click **OK** to close the dialog box. Essbase updates the Data Load dialog box to display the data file you chose.
10. Click **OK** to load the selected file. It should take a minute or two to load, and then the Data Load Completed dialog box appears. Click **Close**.

5.4.3 Preparing the Samppart application

The Samppart application contains one database called Company. To prepare it for use, you need to load data into it. Complete the following steps:

1. Start the Application Manager.
2. Connect to the server in which the sample application is installed.
3. Select the application called **Samppart** from the Applications list box.
4. Select the database called **Company** from the Databases list box.
5. Choose **Database--> Load Data**. The Data Load dialog box appears.
6. Click the **Find** button above the Data files list box to locate a data file to load. The Open Server Data File Objects dialog box appears.
7. Ensure that List Objects of Type is Text files.
8. Select the text file named **Calccomp** from the Objects list box.
9. Click **OK** to close the dialog box. Essbase updates the Data Load dialog box to display the data file you chose.
10. Click **OK** to load the selected file. It should take a minute or two to load, and then the Data Load Completed dialog box appears. Click **Close**.

For the East database of the Sampeast application, choose **Calceast** file in Step 8.

5.4.4 Preparing the Drinks application of Wired for OLAP

The Drinks application contains a database called Demo (which is different from the Demo application of DB2 OLAP Server). After loading data into the Demo database, you should use Wired Analyzer to run the Drinks application of Wired for OLAP. (Of course, prior to loading data into the Drinks application, DB2 OLAP Server and Hyperion Wired for OLAP must be installed and running.)

To install the sample application Drink, follow these steps:

1. Start the Application Manager.

2. Connect to the server in which the sample application is installed, using a valid user ID and password.
3. Select **File--> New--> Application** from the tool bar, and then enter `Drinks` for the application name and click **OK**.
4. Select **File--> New--> Database** from the tool bar, and then enter `Demo` for the database name and click **OK**.
5. Select **Start/Stop** from the Database menu to stop the Demo database.
6. Run the *Wireddirectory\samples\Database* and *Views \Hyperion Drinks Database.exe* to start the extraction of files. You can get Wireddirectory in 5.3.1.3, “Licensed Software - installation steps” on page 118.
7. Select the extraction folder on the PKSFX window; see Figure 66. You store the extracted files to your workstation. In our example, it is C:\Essbase\app\Drinks\Demo.

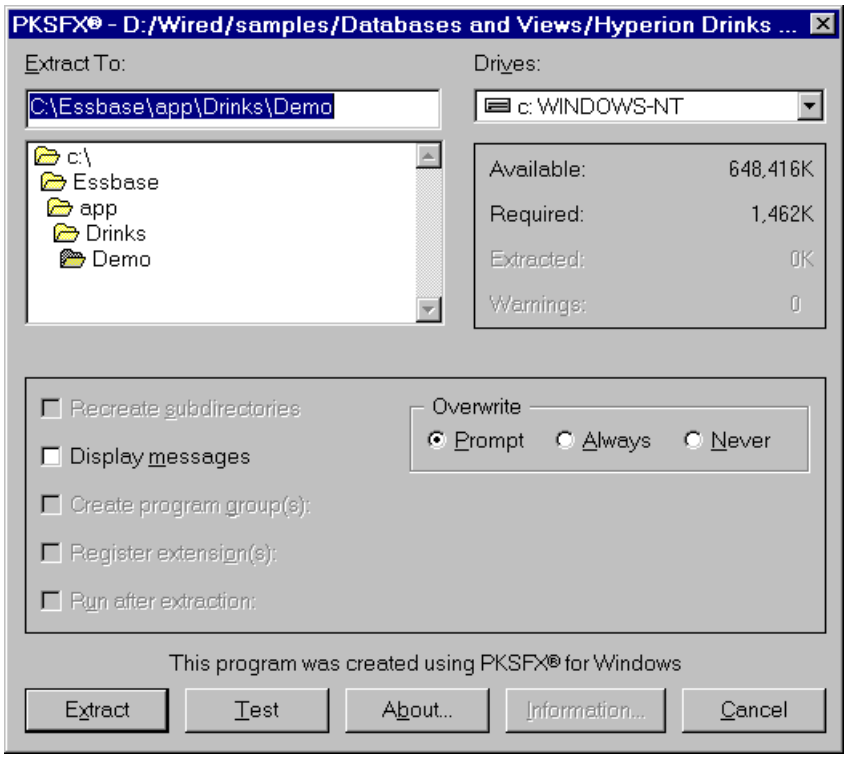


Figure 66. PKSFX window

8. Click the **Extract** button. When prompted to overwrite the existing files, click **Always**. It replaces the Demo.dat and Demo.otl files.
9. When the files are extracted, select **Database--> Start/Stop** to restart the database.
10. Replace the current outline of Demo database that is empty with the new Demo.otl. Select **File--> Open** from Application Manager.
    a. In the Open Client Object screen, select **Server** at location and **File system**.
    b. In the Open Client file screen, go to the directory that the files are extracted to (in our example, it is C:\Essbase\app\Drinks\Demo) and open Demo.otl.
    c. Save the outline file by clicking **File--> Save as**.
    d. In Save server object, click:
       - **Server** at location
       - **Drinks** at application
       - **Demo** at database
       - **Outline** at List Object of type
       - **Demo** at Objects
11. Select **Database--> Load Data**; see Figure 67 on page 139.

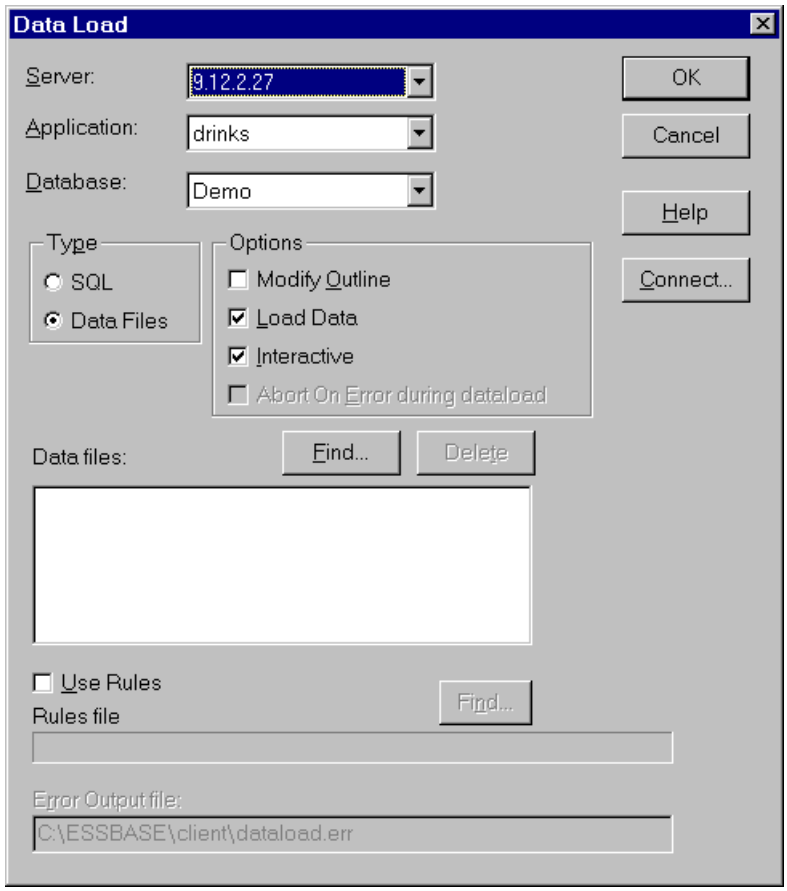


Figure 67. Data Load window

12.In the Data Load window (see Figure 67):

- Select **Data Files** in the Type box
- Select Application and Database name (Drinks, Demo)
- Click the **Find** button.

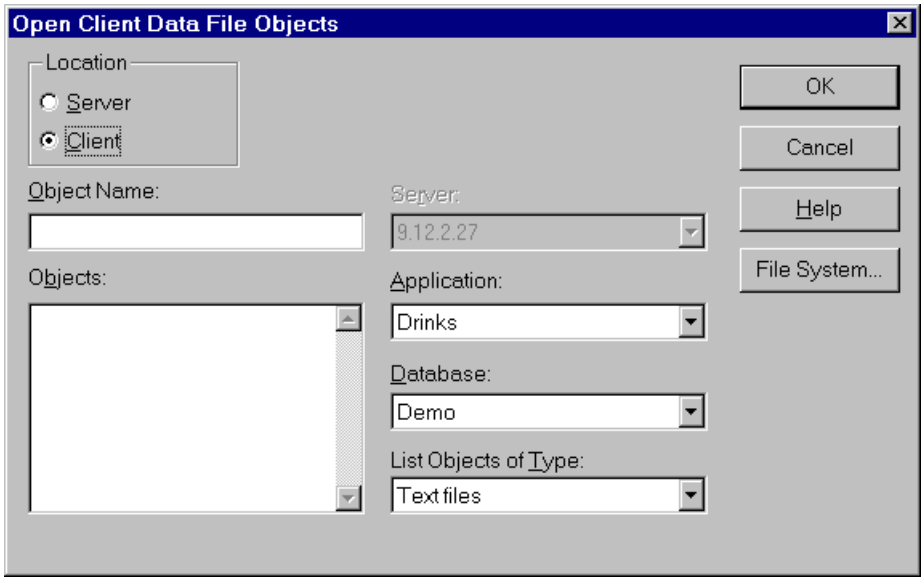


Figure 68. Open Server Data File Objects window

13.In the Open Server Data File Objects window (see Figure 68):

- Select **Client** and **File System**.

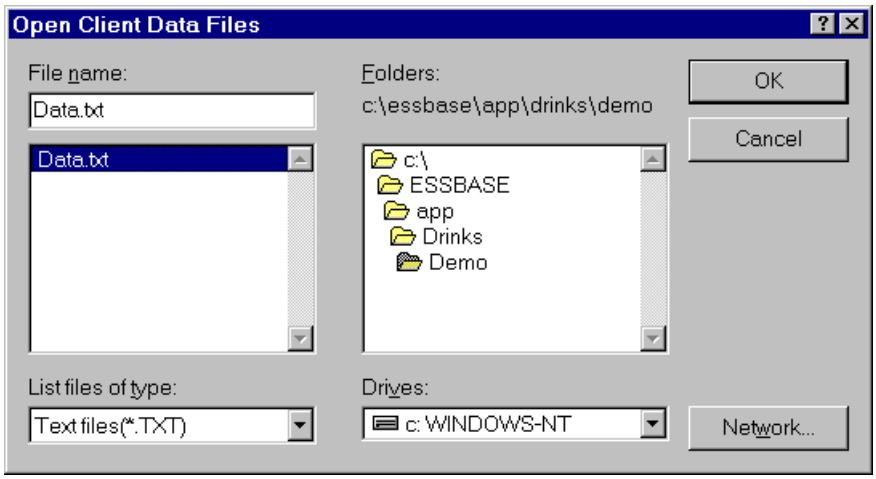


Figure 69. Open Client Data Files window

14. In the Open Client Data Files window (see Figure 69 on page 140):
    - Provide the extraction path (of step 7 and 8).
    - Select **data.txt** from the File name list.
    - Click **OK** to return to the Data Load window.
15. Click **OK** in the Data Load window to start the load process.
16. You should get no error message. Close the Data Load Complete window.
17. Select **Database--> Calculate**.
18. In the Database Calculation window, mark **(Default)** calculation script and click **OK**.
19. Now the application Drinks is ready to use by Wired Analyzer.

5.4.5 Providing user access to the sample applications

Essbase provides a comprehensive security system for a secure multiuser environment. By default, the sample applications are created with a security access level of No Access, which means that no user can connect to the database, unless the user is defined as a supervisor.

The DB2 OLAP Server supervisor, defined during the DB2 OLAP Server installation, automatically has the supervisor privilege. Therefore, the DB2 OLAP Server supervisor can make the sample applications available to other users.

For example, in order to provide all users with *write access* to the database:

1. Start the Essbase Application Manager.
2. Connect to the server using the DB2 OLAP Server supervisor ID.
3. Select the application called **Sample** from the Applications list box.
4. Choose **Application--> Settings**. The Application Settings dialog box appears.
5. Click the **Write** option from the Minimum Database Access group, and click **OK** to save the setting.
6. Select the application called **Demo** from the Applications list box.
7. Choose **Application--> Settings**.
8. Click the **Write** option from the Minimum Database Access group, and click **OK** to save the setting.

Application Sample and Demo are now ready for use. For more details, refer to 9.4, “Administering DB2 OLAP Server security” on page 210.

Chapter 6. Administration of DB2 OLAP Server

This chapter describes the activities needed to administer the DB2 OLAP Server:

- Activate and deactivate the DB2 OLAP Server
- ESSBASE commands
- Backup and recovery of the OLAP database

6.1 Activating the DB2 OLAP Server

To start the DB2 OLAP Server, you can choose one of the following three methods:

- In foreground mode
- In background mode
- Using a started task procedure

Based on the way DB2 OLAP Server has been started, you can choose a way to deactivate it.

If DB2 OLAP Server is not activated properly, check if there is any active process that remains active even after the DB2 OLAP Server is deactivated.

6.1.1 Starting the DB2 OLAP Server in foreground mode

To start the DB2 OLAP Server engine from the OMVS command line or telnet session, enter:

```
ESSBASE password
```

where *password* is the password of the supervisor. This starts the DB2 OLAP Server process in the OS/390 UNIX environment. You will get the following messages:

```
Startup sequence completed
Security is enabled
Logins are enabled
Waiting for Client Requests...
```

From this session, you can use the following commands:

- START *application* - start an application
- STOP *application* - stop an application
- USERS - list all connected users
- PORTS - list port usage
- LOGOUTUSER *user* - forcibly logout a user
- PASSWORD - change the Essbase system password
- DUMP *filename* - dump the current state of security to a file
- VERSION - display the Essbase version number
- HELP - display the help text
- QUIT/EXIT - exit program (stop all applications)

6.1.2 Starting the DB2 OLAP Server in background mode

To start the DB2 OLAP Server engine in background mode from the OMVS command line or telnet session, enter:

```
ESSBASE password -b&
```

where *password* is the password of the supervisor. DB2 OLAP Server starts in the background and the OMVS command prompt is returned. You can check if OLAP Server engine is active by using:

```
ps -ef
```

The active process will look like:

```
587202585  637534252  - 15:15:44 ? 0:07 ESSBASE password -b
```

6.1.3 Starting DB2 OLAP Server using a started task procedure

You can create an MVS started task procedure to start the DB2 OLAP Server as follows:

1. Create a script file start.sh (in our example, in /u/essbase/rsm) shown in Figure 70. The permission bits of start.sh should be set to 755.

```
cd /u/olapr
. .profile
cd /u/essbase/rsm/bin
ESSBASE password -b
```

Figure 70. Start.sh script for DB2 OLAP Server

In the start.sh script, you have to:

a. Specify the directory where the .profile of the supervisor resides. In our example, it is /u/olapr.

b. Run the .profile.

c. Specify the directory where the ESSBASE command resides. In our example, it is /u/essbase/rsm/bin.

d. Type the ESSBASE command with the supervisor's password.

The DB2 OLAP Server will start as a background process.

2. Create a new member in your SYS1.PROCLIB data set, as shown Figure 71.

```
//DB2OLAPR PROC
//*****************************************************************
//*
//* PROCEDURE TO START DB2 OLAP SERVER RELATIONAL AS AN STC
//*
//*****************************************************************
//DB2OLAPR EXEC PGM=BPXBATCH,REGION=0M,TIME=1440,
//             PARM='sh  /u/essbase/rsm/start.sh'
//STDOUT   DD  PATH='/u/essbase/rsm/BPXBATCH.OUT',
//             PATHMODE=SIRWXU,
//             PATHOPTS=(OWRONLY,OCREAT,OTRUNC)
//STDERR   DD  PATH='/u/essbase/rsm/BPXBATCH.ERR',
//             PATHMODE=SIRWXU,
//             PATHOPTS=(OWRONLY,OCREAT,OTRUNC)
//SYSPRINT DD  SYSOUT=*
//CEEDUMP  DD  SYSOUT=*
//STDENV   DD  DUMMY
```

Figure 71. Started task procedure for DB2 OLAP Server relational

- DB2OLAPR is the started task procedure name.
- PARM contains a shell command to execute a shell script.
- STDOUT is the output file of the DB2OLAPR.
- STDERR is the error log file of the DB2OLAPR.

3. Enable the MVS started task (DB2OLAPR) to execute under the DB2 OLAP Server supervisor ID (OLAPR) by running the sample JCL shown in Figure 72 on page 146. Contact your security administrator for more details.

.

```
//BI390B JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=BI390B
//*************************************************************
//IKJEFT   EXEC PGM=IKJEFT01,REGION=3072K
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
  RDEFINE STARTED DB2OLAPR.** STDATA(USER(OLAPR))
  SETROPTS RACLIST(STARTED) REFRESH
/*
```

Figure 72. RACF definitions for started task procedure

4. Start the DB2 OLAP Server as an MVS started procedure from MVS SYSLOG:

   /s DB2OLAPR

Figure 73 shows the informational message at the SYSLOG, when you start DB2 OLAP Server using a started task.

```
S DB2OLAPR
$HASP100 DB2OLAPR ON STCINRDR
IEF695I START DB2OLAPR WITH JOBNAME DB2OLAPR IS ASSIGNED TO USER
OLAPR   , GROUP SYS1
$HASP373 DB2OLAPR STARTED
IEF403I DB2OLAPR - STARTED - TIME=14.04.09 - ASID=0085.
-                                          --TIMINGS (MINS.)--
   ----PAGING COUNTS---
-JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK   SERV
PG   PAGE   SWAP    VIO SWAPS
-DB2OLAPR DB2OLAPR DB2OLAPR    00     53    .00    .00     .0   1066
 0      0      0      0     0
-                                          --TIMINGS (MINS.)--
   ----PAGING COUNTS---
-JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK   SERV
PG   PAGE   SWAP    VIO SWAPS
```

Figure 73. SYSLOG after activating DB2 OLAP Server using a started task

If the DB2 OLAP Server process is not active, look at BPXBATCH.OUT (STDOUT) or BPXBATCH.ERR (STDERR) to determine why; see Figure 71 on page 145.

6.2 Deactivating the DB2 OLAP Server

There are three ways to shut down the OLAP server engine:

- Using the EXIT command
- Using ESSCMD interactively
- Using a started task procedure with an ESSCMD batch script

6.2.1 Using the EXIT command

When you start the DB2 OLAP Server in foreground mode (using OMVS or telnet), the session is waiting for your input and becomes a log of DB2 OLAP Server. You can shut down the DB2 OLAP Server by entering:

```
EXIT or QUIT
```

Even if you close this session without entering the above command, DB2 OLAP Server will be shut down.

6.2.2 Using ESSCMD interactively

When you shut down the DB2 OLAP Server from ESSCMD, you must have the supervisor privilege to use this command. From an OMVS session or a telnet session, enter ESSCMD, then provide:

- SHUTDOWNSERVER
- Server name or IP address
- User ID and password
- EXIT

6.2.3 Using a started task procedure with the ESSCMD batch script

Another way to shut down the DB2 OLAP Server is by using a batch script, in just the same way as you start it. (See 6.1.3, “Starting DB2 OLAP Server using a started task procedure” on page 144.)

1. Create a script file stop.sh (in our example, in the /u/essbase/rsm directory), as shown in Figure 74 on page 148 and create a script file stop.scr as shown in Figure 75 on page 148. The permission bits of these two members should be set to 755.

```
cd /u/olapr
. .profile
cd /u/essbase/rsm/
ESSCMD stop.scr
```

Figure 74. Sample stop.sh script

```
SHUTDOWNSERVER  "9.12.2.26" "olapr" "password"  ;
   Exit;
```

Figure 75. Sample stop.scr file

- /u/olapr is the home directory of the supervisor.
- .profile is the profile of the supervisor.
- 9.12.2.26 is the IP address.
- *olapr* is the supervisor ID.
- *password* is the password of the supervisor.

2. Create a new member in your SYS1.PROCLIB data set as shown in Figure 76.

```
//STOPRSM  PROC
//*****************************************************
//*
//* STC PROCEDURE TO STOP DB2 OLAP SERVER RELATIONAL
//*
//*****************************************************
//STOPRSM  EXEC PGM=BPXBATCH,REGION=0M,TIME=1440,
//             PARM='sh  /u/essbase/rsm/stop.sh'
//STDOUT   DD  PATH='/u/essbase/rsm/stop.out',
//             PATHMODE=SIRWXU,
//             PATHOPTS=(OWRONLY,OCREAT,OTRUNC)
//STDERR   DD  PATH='/u/essbase/rsm/stop.err',
//             PATHMODE=SIRWXU,
//             PATHOPTS=(OWRONLY,OCREAT,OTRUNC)
//SYSPRINT DD  SYSOUT=*
//CEEDUMP  DD  SYSOUT=*
//STDENV   DD  DUMMY
```

Figure 76. Sample started task to shut down the DB2 OLAP Server

3. Enable the MVS started task (STOPRSM) to execute under the DB2 OLAP Server supervisor ID (OLAPR) by running the JCL shown in Figure 77, or ask your security administrator for assistance.

```
//BI390B  JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=BI390B
//*************************************************************
//IKJEFT   EXEC PGM=IKJEFT01,REGION=3072K
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
  RDEFINE STARTED STOPRSM.** STDATA(USER(OLAPR))
  SETROPTS RACLIST(STARTED) REFRESH
/*
```

Figure 77. RACF definition for running a started task

- STOPRSM is the MVS started task procedure name.
- OLAPR is the DB2 OLAP Server supervisor ID.

4. Stop the DB2 OLAP Server by invoking an MVS started procedure from MVS SYSLOG:

```
/s STOPRSM
```

Figure 78 on page 150 shows the informational message at the SYSLOG, when you start DB2 OLAP Server using a started task. You should verify the return code 0.

```
S STOPRSM
$HASP100 STOPRSM  ON STCINRDR
IEF695I START STOPRSM  WITH JOBNAME STOPRSM  IS ASSIGNED TO USER STC
   , GROUP SYS1
$HASP373 STOPRSM  STARTED
IEF403I STOPRSM - STARTED - TIME=14.16.48 - ASID=0085.
-                                          --TIMINGS (MINS.)--
   ----PAGING COUNTS---
-JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK   SERV
PG   PAGE   SWAP    VIO SWAPS
-STOPRSM  STOPRSM  STOPRSM     00     50    .00    .00     .0   1003
 0      0      0      0     0
-                                          --TIMINGS (MINS.)--
   ----PAGING COUNTS---
-JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK   SERV
PG   PAGE   SWAP    VIO SWAPS
-STOPRSM  *OMVSEX  STOPRSM     00     14    .00    .00     .0    925
 0      0      0      0     0
```

Figure 78. SYSLOG after shutting down DB2 OLAP Server using a started task

If the DB2 OLAP Server process is still active, look at STOP.OUT (STDOUT) or STOP.ERR (STDERR) to determine why. (See Figure 76 on page 148.)

6.3 Administering the Essbase log file

The Essbase.log file is an ever-growing file and if you do not monitor it regularly, your OLAP HFS will eventually fill up. This situation causes the DB2 OLAP Server to hang. Therefore, you should monitor this log file, and either move it to another HFS file or delete it whenever it is too large. This file is located in the DB2 OLAP Server home directory.

Attention

When you get TCP/IP errors during installation or test, your Essbase.log file will grow particularly fast, causing the HFS file to fill up.

In this case, you should cancel the task that is causing the problem by pressing Shift+6 and `c`, and then handle the log file as previously described.

6.4 ESSCMD: ESSBASE command

ESSCMD is a command line interface that performs operations interactively or through a batch or script file.

Generally, when you are performing simple tasks that require few commands, interactive mode is preferred. If you are performing more complex tasks that require many commands, consider using a script file or a batch file.

This section describes how to use ESSCMD in each environment. For a complete list of Essbase commands, refer to *Essbase Quick Technical Reference*, SC26-9239.

6.4.1 Using interactive mode

You can run Essbase commands from an OS/390 UNIX prompt either from an OMVS session or a telnet session. This is useful when you are performing simple tasks that require few commands.

In the following example, we load the Demo/ Basic application using Essbase commands in interactive mode; see Figure 79 on page 152.

1. From the UNIX prompt, issue the command:

   ```
   ESSCMD
   ```

2. Login to the server.
3. Enter the server host name or IP address, your Essbase user ID, and password. The server gives you a list of applications available for your user ID.
4. Select the application you want to load.
5. Enter the Essbase load command.
6. Then, enter other Essbase commands or quit the dialog.

```
BI390B:/u/bi390b: >ESSCMD
Essbase Command Mode Interface - 5.0.2 Patch2 (03/19/99)
Copyright(c) 1991-1999 Hyperion Solutions Corporation
U.S. Patent Number 5,359,724
All rights reserved
:::Ý0¨->login
Login:
Host Node >9.12.2.26
User >olapm
Password >olap390
ÝWed Dec  1 15:25:30 1999¨Local////Info(1051034)
Logging in user olapm
ÝWed Dec  1 15:25:30 1999¨Local////Info(1051035)
Last login on Wed Dec  1 11:05:22 1999
ÝWed Dec  1 15:25:30 1999¨Local////Info(1051036)
0 unsuccessful attempt(s) since last login
Available Application Database:
Demo     ==> Basic
Sample   ==> Basic
Sample   ==> Interntl
. . .
. . .
9.12.2.26:::olapmÝ1¨->select Demo Basic
Select:
ÝWed Dec  1 19:35:32 1999¨9.12.2.26///olapm/Info(1054014)
Database Basic loaded
ÝWed Dec  1 19:35:32 1999¨9.12.2.26///olapm/Info(1051061)
Application Demo loaded - connection established
9.12.2.26:Demo:Basic:olapmÝ1¨->
Loaddata 3 "/u/essbase/mdsm/app/Demo/Basic/Data.txt"
LoadData:

ÝWed Dec  1 15:30:37 1999¨9.12.2.26/Demo/Basic/olapm/Info(1003024)
Data Load Elapsed Time : Ý9.22¨ seconds

sts = 0 DataFile = /u/essbase/mdsm/app/Demo/Basic/Data.txt

9.12.2.26:Demo:Basic:olapmÝ1¨->exit
Exit:
Wed Dec  1 15:32:19 1999¨9.12.2.26/Demo/Basic/olapm/Info(1051037)
Logging out user olapm, active for 6 minutes
BI390B:/u/bi390b: >
```

Figure 79. Sample Essbase commands using interactive mode

6.4.2 Using batch mode

This section provides an example of how to run Essbase commands such as load, calc, and so on, in an OS/390 environment using batch jobs.

Figure 80 shows how to run the DB2 OLAP Server load script in batch by invoking the BPXBATCH program. This example assumes that the DB2 OLAP Server is already running.

1. Create a JCL that invokes the BPXBATCH program.

```
//BI390B  JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=BI390B
//*************************************************************
//* SAMPLE JCL TO LOAD SAMPLE DEMO APPLICATION
//*************************************************************
//LOAD     EXEC PGM=BPXBATCH,REGION=256M,
//     PARM='sh  /u/essbase/rsm/load.sh'
//STDOUT   DD  PATH='/u/essbase/rsm/load.out',
//             PATHMODE=SIRWXU,
//             PATHOPTS=(OWRONLY,OCREAT,OTRUNC)
//STDERR   DD  PATH='/u/essbase/rsm/load.err',
//             PATHMODE=SIRWXU,
//             PATHOPTS=(OWRONLY,OCREAT,OTRUNC)
//SYSUDUMP DD  DUMMY
//SUMDUMP  DD  DUMMY
//SYSPRINT DD  SYSOUT=*
//STDENV   DD  DUMMY
```

Figure 80. Sample JCL to run a load Essbase command

When submitted, this JCL invokes the load.sh script which invokes the load.scr script to run the load operation (see Figure 81). These scripts reside in the /u/essbase/rsm directory. The output files for this operation are located in the files you specify in the STDOUT and STDERR DD statements (load.out for the outfile and load.err for the error file).

2. Create the script (load.sh) that invokes the load.scr script.

```
cd /u/olapm
. .profile
cd /u/essbase/rsm
ESSCMD load.scr
```

Figure 81. Sample load.sh script to invoke the Essbase load command

The .profile file is the user profile that will run the batch job, and it should include the environment variables to process ESSCMD correctly. Then, change directory to where the load script is located.

3. Create the load script that invokes the `loaddata` command; see Figure 82.

```
Login "9.12.2.26" "olapr" "password"
Select "Demo" "Basic" ;
Resetdb ;
Loaddata 3 "/u/essbase/rsm/app/Demo/Basic/Data.txt" ;
  Exit ;
```

Figure 82. Sample load.scr script containing the Essbase commands

In the load.scr script, you have to:

- Login to the DB2 OLAP Server using an authorized user ID. We used *9.12.2.26* as the server IP address, *olapr* as a user ID that runs this batch job, and *password* as a password of the olapr.
- Select the application database you want to load.
- Reset the database.
- Load the data, specifying the path to where the source data file is stored.

For a complete list of Essbase commands, refer to *Essbase Quick Technical Reference*, SC26-9239.

6.5 Backup and recovery of the OLAP database

A key part of database maintenance is making regular backups of the data and then, if the database is corrupted or damaged, recovering it at the time of the last successful commit.

With the DB2 OLAP Server, when the database is crashed you can reload and recalculate data using source data. Depending on the data volume, it could be a short process. With a large volume of data and many dimensions, rerunning those jobs takes longer to rebuild the database. You should consider how to back up and recover your large OLAP databases.

This section describes how to backup a database, how to recover a database, and other ways DB2 OLAP Server protects your data. You have chosen either MDSM or RSM to store the OLAP data at installation time. Based on this

choice, you should develop your own procedures for backing up and recovering the OLAP database.

In addition, you have to back up the following files at the same time that you are backing up databases:

- database name.*otl*: Database outline file
- database name.*esm*: a main root page file
- database name.*tct:* a transaction control table
- database name.*csc*: a calculation script
- database name.*rul*: a rule file

6.5.1 Back up and recover the MDSM database

You can use either OS/390 tools or original Hyperion backup tools to back up files under MDSM. We focus on OS/390 tools because they are usually faster than Hyperion backup tools, and because most installations already have OS/390 skills. For example, if you are familiar with DFSMSdss, which is a typical MVS file management tool, you can use it for your entire file backup. The tools we describe in this section are:

- ADSTAR Distributed Storage Manager (ADSM)
- DFSMSdss
- TAR command
- ARCHIVING (Hyperion)
- EXPORT (Hyperion)

Depending on your tool selection, you need to track all backup activities. In the case of ADSM, it keeps all backup information for recovery.

6.5.1.1 Using ADSM

If ADSM has already been implemented as a company-wide backup and recovery process, you can also use it for your OLAP database and files. Tivoli ADSTAR Distributed Storage Manager (ADSM) is IBM's solution for distributed data management in an enterprise network environment. ADSM supports a wide variety of platforms for both small and large systems. It also delivers many data management functions, such as data backup, data archival, space management, and disaster recovery integrated in one software solution. For more information, refer to *Getting Started with ADSM: A Practical Implementation Guide*, SG24-5416.

Using ADSM, you can maintain backup versions of HFS files that can be restored quickly and easily if the original files are damaged or lost. You can also archive files that are not currently needed on an HFS system, and retrieve them when necessary.

There are two options to choose at backup:

- Incremental: This option creates backups for all members in the directory and/or subdirectory when it is first used and then creates incremental copies of members that changed after the last backup.
- Selective: This option creates backups for selected members only.

To start an incremental backup from an OMVS session, use the following command:

```
dsmc incremental /home/subdirectory/ -subdir=yes
```

Figure 83 on page 157 shows the information messages you will receive after issuing the ADSM command.

Before starting to restore files, you should be aware of the policies that the ADSM Sever follows about keeping backup versions:

- ADSM creates a new backup only if you change the files. Otherwise, no new backup version will be created.
- The ADSM Server keeps the newest version of the backup as long as you have it in your storage. It will be kept for as many days as specified in your definition after you delete the original files.
- ADSM keeps several recent backup versions, based on your definition, and you can choose one of them when you restore.

```
BI390A:/u: >dsmc incremental /u/essbase/app/Samppart/ -subdir=yes
ADSTAR Distributed Storage Manager
Command Line Backup Client Interface - Version 3, Release 1, Level 0.7
(C) Copyright IBM Corporation, 1990, 1999, All Rights Reserved.

Node Name: WTSC66OE
Session established with server ADSM: MVS
Server Version 3, Release 1, Level 2.40
Server date/time: 12/07/1999 16:44:11  Last access: 12/07/1999 16:24:22

Incremental backup of volume '/u/essbase/app/Samppart/'
Directory--> 8,192 /u/essbase/app/Samppart [Sent]
Directory--> 8,192 /u/essbase/app/Samppart/Company[Sent]
Normal File--> 155 /u/essbase/app/Samppart/Samppart.apb[Sent]
Normal File--> 155 /u/essbase/app/Samppart/Samppart.app [Sent]
Normal File--> 19,397 /u/essbase/app/Samppart/Samppart.log
Normal File--> 1,027,072 /u/essbase/app/Samppart/Company/ess00001.ind
Normal File--> 2,140,744 /u/essbase/app/Samppart/Company/ess00001.pag
......
Normal File--> 132 /u/essbase/app/Samppart/Company/Company.tct
Successful incremental backup of '/u/essbase/app/Samppart/'

Total number of objects inspected:      15
Total number of objects backed up:      15
Total number of objects updated:         0
Total number of objects rebound:         0
Total number of objects deleted:         0
Total number of objects failed:          0
Total number of bytes transferred:    4.24 MB
Data transfer time:                  6.62 sec
Network data transfer rate:        656.40 KB/sec
Aggregate data transfer rate:      123.10 KB/sec
Objects compressed by:                 0%
Elapsed processing time:           00:00:35
```

Figure 83. ADSM command to back up an OLAP database directory

Figure 84 on page 158 shows an example of restoring files using the ADSM command:

```
dsmc restore /u/essbase/app/Samppart/ -subdir=yes
```

```
BI390A:/u: >dsmc restore /u/essbase/app/Samppart/ -subdir=yes
ADSTAR Distributed Storage Manager
Command Line Backup Client Interface - Version 3, Release 1, Level 0.7
(C) Copyright IBM Corporation, 1990, 1999, All Rights Reserved.

Restore function invoked.

Node Name: WTSC66OE
Session established with server ADSM: MVS
Server Version 3, Release 1, Level 2.40
Server date/time: 12/07/1999 17:17:59 Last access: 12/07/1999 17:16:00

ANS1247I Waiting for files from the server...
Restoring 8,192 /u/essbase/app/Samppart/Company

--- User Action is Required ---
File '/u/essbase/app/Samppart/Samppart.apb' exists

Select an appropriate action
  1. Replace this object
  2. Replace all objects that already exist
  3. Skip this object
  4. Skip all objects that already exist
  A. Abort this operation
Action [1,2,3,4,A] : 1
Restoring 155 /u/essbase/app/Samppart/Samppart.apb
Restoring 144 /u/essbase/app/Samppart/Company/Company.ind
  ......................
Restoring 4,853 /u/essbase/app/Samppart/Company/Company.otl
Restoring 132 /u/essbase/app/Samppart/Company/Company.tct

Restore processing finished.

Total number of objects restored:       14
Total number of objects failed:          0
Total number of bytes transferred:    4.24 MB
Data transfer time:                   6.19 sec
Network data transfer rate:         702.58 KB/sec
Aggregate data transfer rate:       134.46 KB/sec
Elapsed processing time:            00:00:32
```

Figure 84. ADSM command to restore directory and members

6.5.1.2 Using DFSMSdss

You can use DFSMSdss for both MDSM and RSM. Note, however, that with DFSMSdss, you can only copy the entire HFS file system, not specific members or directories.

```
//BKUPHFS9 JOB (999,POK),'L06R',CLASS=A,REGION=4096K,
//             MSGCLASS=T,TIME=10,MSGLEVEL=(1,1),NOTIFY=&SYSUID
/*JOBPARM L=999,SYSAFF=SC62
//*************************************************************
//* DUMP (LOGICAL)  HFS
//* FILE SYSTEM NAME:  FILE SYSTEM OF MDSM
//*
//* THIS JOB WILL DUMP FILE SYSTEM FOR DB2 OLAP SERVER
//* DATA SETS WILL RESIDE ON THE DISK PACK INDICATED BY HFSOUT
//*************************************************************
//BACKUP   EXEC  PGM=ADRDSSU
//SYSPRINT DD    SYSOUT=*
//HFS1     DD    UNIT=3390,VOL=SER=TSMS25,DISP=SHR
//HFSOUT   DD    DSN=BI390B.SC62.MDSM.SEQ,
//            DISP=(NEW,CATLG,DELETE),SPACE=(CYL,(50,25),RLSE),
//           UNIT=3390,VOL=SER=TOTTS1
//SYSIN    DD *
 DUMP DATASET(INCLUDE(OMVS.SC62.MDSM)) -
       COMPRESS TOL(ENQF) -
       LOGINDDNAME(HFS1) OUTDDNAME(HFSOUT) ALLDATA(*) ALLEXCP
/*
```

Figure 85. Dump entire HFS file system using DFSMSdss

You can also dump the whole volume which contains multiple HFS files by specifying the volume serial number.

When you restore an HFS file from the DFSMSdss backup, you should use DFSMSdss. Figure 86 on page 160 shows what the job looks like.

```
//RESTHFS  JOB (999,POK),'HFS RESTORE',CLASS=A,REGION=4M,
//             MSGCLASS=T,TIME=10,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//*************************************************************
//* THIS JOB WILL RESTORE THE PDSE FILES FROM THE SEQUENTIAL
//* BACKUP.
//* FILE SYSTEM NAME:  OMVS.SC62.MDSM
//*************************************************************
//ROOT     EXEC PGM=ADRDSSU,REGION=6M
//SYSPRINT DD   SYSOUT=*
//SEQ1   DD DSN=BI390B.SC62.MDSM.SEQ,DISP=SHR
//DASD1    DD   UNIT=3390,VOL=SER=TSMS29,DISP=SHR
//SYSIN    DD   *
 RESTORE  INDD(SEQ1) OUTDD(DASD1) TOL(ENQF) -
 DATASET(INCLUDE(**))          -
 STORCLAS(SCCOMP)             -
 CANCELERROR
/*
```

Figure 86. Restore a HFS file using DFSMSdss

Note that you do not select a specific directory to restore; you can only restore the entire HFS file system.

6.5.1.3 Using the tar command

If you are familiar with the UNIX environment, you can use the *tar* command to back up and restore. With options of the *tar* command you can copy all members in a directory and restore all or some of them.

For backup, go to the directory you want to back up, and then enter the following commands:

```
cd /u/essbase/
tar -cvf /tmp/olap.tar
```

In these commands:

- /u/essbase is the directory you want to back up.
- /tmp is the target directory and olap.tar is the name of backup file.

You should take note of the date, the directory, and members copied into the tar file.

For restore, go to the directory where you want to restore files, and then enter the following commands:

```
cd /u/new/essbase/
tar -xvf /tmp/olap.tar
```

where /u/new/essbase is the target directory where you want to restore files.

You can select the members you want to restore by using the following command:

```
tar -xvf /tmp/olap.tar/member-name
```

6.5.1.4 Using archiving

To use the archiving function provided by Hyperion to archive your data, you should follow these three steps:

1. Issue the BEGINARCHIVE command in ESSCMD.

   BEGINARCHIVE does the following:

   a. Commits any modified data blocks.

   b. Switches the database into read-only mode.

   c. Creates a file containing a list of files that need to be backed up. By default, the file is called archive.lst.

2. Use a utility to back up the files listed in archive.lst. You can use methods such as ADSM, DFSMSdss, and the tar command, all of which are described in previous sections.

   Be sure to back up data on every disk volume Essbase uses. For information about data storage on multiple volumes, see *DB2 OLAP Server DB Administrator's Guide*, SC26-9286.

3. Complete the archiving process by issuing the ENDARCHIVE command in ESSCMD.

   ENDARCHIVE does the following:

   a. It puts the database into read-write mode.

   b. It reopens database files in exclusive mode.

Only a media failure (faulty disk, disk failure, or head crash) would require you to restore from archived files. For more information, refer to *DB2 OLAP Server Administrator's Guide*, SC26-9286.

6.5.1.5 Using exporting

You can back up data by exporting instead of archiving. The EXPORT utility copies data to an ASCII text file that you specify; it does not compress the data. The exported file contains data only and does not include control, outline, or security information.

You might use the EXPORT utility when:

- Transferring data across platforms.
- Backing up only a certain portion of the data; for example, level 0 blocks.
- You prefer an exported file in text format, rather than binary format.

To export data, choose **Database--> Export** in the Application Manager or issue EXPORT in ESSCMD. Figure 87 on page 163 shows the panel of the Application Manager you use to specify how the data can be exported.

Enter the following values:

- Server file name: the name of the exported file with a *.txt* extension and UNIX system directory where you keep it. By default, the exported file is kept at the /essbase home/app directory. In our example, the exported file name is vlsiback.txt and it will be kept in the /u/olapr directory.
- Options:
  - All data
  - Level 0 blocks only (blocks containing only Level 0 sparse member combinations)
  - Data from input blocks only (blocks containing data from a data load)

You can reload data from the exported file by selecting **Database--> Load data** from the Application Manager. If Server is selected, the object to load must reside in the database directory under */essbase-home/app/application_name/database_name*. Type the name of the object in the Object Name text box, or select it from the Objects list box.

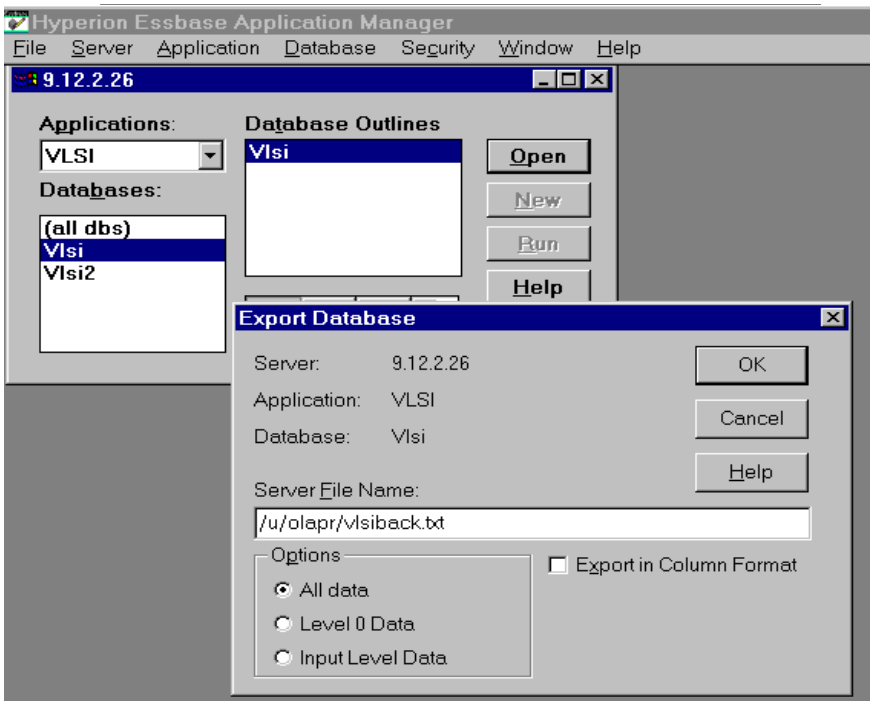


Figure 87. Exporting data

When you reload all data in a database from the exported files, Essbase marks all blocks in the database as input blocks. You do not have to recalculate it. But if you reload data exported from Level 0 blocks or input blocks, you must recalculate the database after reloading. For detailed information on how to do this, refer to *DB2 OLAP Server Administrator's Guide*, SC26-9286.

6.5.2 Back up and recover RSM database

When you use the relational storage manager and want to make a backup copy of the OLAP database, you can use the DB2 COPY utility in addition to volume backup or data set backup. In this section, we describe:

- Using DB2 utilities
- Using DFSMSdss

For the outline file, calculation script, rule file, or main root page file, you can use the methods described in 6.5.1, "Back up and recover the MDSM database" on page 155.

6.5.2.1 Using DB2 utilities

You can use the DB2 COPY utility to make a backup copy of the tablespace and the indexes. You create your own DB2 database and tablespaces for each OLAP application, so you know the database name and tablespace name. However, because many indexes are built when you load data into the database, and their names are created by DB2 OLAP Server, you can get index names by retrieving DB2 catalog tables.

You can make a backup of a tablespace only or of a tablespace and its indexes, respectively. When you have a backup copy of the tablespaces only, you should recover the tablespace first, and then rebuild its indexes using the REBUILD INDEX utility (DB2 Version 6). In DB2 Version 5, you should run the RECOVER INDEX utility to rebuild indexes.

If you have a backup copy of a tablespace and its indexes, you can recover the tablespace using RECOVER TABLESPACE and the indexes using RECOVER INDEX utility (DB2 Version 6 only). Making a backup for the indexes reduces the elapsed time of recovery, especially in case of *large* databases. We recommend you use DB2 COPY utilities for backing up both tablespaces and indexes.

Figure 88 shows a job which makes a backup of the OLAP tablespace using the DB2 COPY utility.

```
//BI390B  JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*      BACKUP OLAP SAMPLE DATABASE
//        EXEC DSNUPROC,PARM='DBH1,IC'
//STEPLIB  DD  DSN=DB2V61H1.SDSNEXIT,DISP=SHR
//         DD  DSN=DSN610.SDSNLOAD,DISP=SHR
//SYSCOPY  DD  UNIT=SYSDA,SPACE=(CYL,(10,2),,,ROUND),
//     DSN=BI390B.BACKUP.OLAPTS,DISP=(NEW,CATLG,CATLG),VOL=SER=TOTDBK
//SYSIN    DD  *
  COPY TABLESPACE  OLAPDB.OLAPTS  SHRLEVEL REFERENCE FULL YES
/*
```

Figure 88. Sample DB2 COPY utility for Sample database

If you want to make a backup copy of a DB2 index, the *copy yes* attribute for each index should be defined. DB2 OLAP Server creates each index without this attribute. Therefore, as shown Figure 89 on page 165, you have to execute ALTER INDEX for every index before you make a copy of it.

```
//ALTER JOB (999,POK),'DB2V510P',CLASS=A,MSGCLASS=T,
// NOTIFY=BI390B,TIME=1440,REGION=4M
/*JOBPARM L=999,SYSAFF=SC62
//JOBLIB   DD  DSN=DB2V61H1.SDSNEXIT,DISP=SHR
//         DD  DSN=DSN610.SDSNLOAD,DISP=SHR
//SQL      EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSTSIN  DD  *
 DSN SYSTEM(DBH1)
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIA61) -
      LIB('DB2V61H1.RUNLIB.LOAD')
 END
//*
//SYSIN    DD *
  ALTER INDEX  OLAPR.CUBE1DINDEX1 COPY YES ;
  ALTER INDEX  OLAPR.CUBE1DINDEX2 COPY YES ;
  ALTER INDEX  OLAPR.CUBE1DINDEX3 COPY YES ;
-- ADD YOUR INDEX NAMES, HERE
//*
```

Figure 89. Sample JCL of altering index attribute

Figure 90 on page 166 shows sample JCL to use when copying indexes using the DB2 COPY utility.

```
//BI390B  JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*      BACKUP INDEXES OF OLAP SAMPLE DATABASE
/*JOBPARM SYSAFF=SC62
//        EXEC DSNUPROC,PARM='DBH1,IC'
//STEPLIB  DD  DSN=DB2V61H1.SDSNEXIT,DISP=SHR
//         DD  DSN=DSN610.SDSNLOAD,DISP=SHR
//SYSCOPY  DD  UNIT=SYSDA,SPACE=(CYL,(1,2),,,ROUND),
//    DSN=BI390B.BACKUP.OLAPTS.I0,DISP=(NEW,CATLG,CATLG),VOL=SER=TOTDBK
//SYSCOP1  DD  UNIT=SYSDA,SPACE=(CYL,(1,2),,,ROUND),
//    DSN=BI390B.BACKUP.OLAPTS.I1,DISP=(NEW,CATLG,CATLG),VOL=SER=TOTDBK
//SYSCOP2  DD  UNIT=SYSDA,SPACE=(CYL,(1,2),,,ROUND),
//    DSN=BI390B.BACKUP.OLAPTS.I2,DISP=(NEW,CATLG,CATLG),VOL=SER=TOTDBK
//*
//*  ADD MORE DATASETS FOR OTHER INDEXES HERE
//*
//SYSIN    DD  *
  COPY INDEX  OLAPR.CUBE1DINDEX1  COPYDDN  SYSCOPY
  COPY INDEX  OLAPR.CUBE1DINDEX2  COPYDDN  SYSCOP1
  COPY INDEX  OLAPR.CUBE1DINDEX3  COPYDDN  SYSCOP2
  ..............................
  ADD MORE COPY INDEX STATEMENTS HERE
/*
```

Figure 90. Sample JCL for backup of the indexes of OLAP sample database

You should have a SYSCOPx data set for each index.

When you recover the database due to media failure or a *broken* database, you use the following steps.

With DB2 Version 6:

1. Recover the tablespace using the RECOVER TABLESPACE utility (see Figure 91 on page 167).
2. Recover indexes with their backup copies (see Figure 92 on page 167) or rebuild the indexes using the REBUILD INDEX utility. We recommend you use RECOVER INDEX for a large database using a backed-up copy.

   DB2 always uses the latest backup file, so you do not have to specify an input file when you recover a tablespace or an index. For more details, refer to *DB2 UDB for OS/390 Utility Guide and Reference*, SC26-9015.

```
//BI390B JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*      RECOVER OLAP SAMPLE DATABASE
/*JOBPARM SYSAFF=SC62
//        EXEC DSNUPROC,PARM='DBH1,RECOVTS'
//STEPLIB  DD  DSN=DB2V61H1.SDSNEXIT,DISP=SHR
//         DD  DSN=DSN610.SDSNLOAD,DISP=SHR
//DSNTRACE DD  SYSOUT=*
//SYSIN    DD  *
  RECOVER TABLESPACE  OLAPDB.OLAPTS
/*
```

Figure 91. Sample JCL for recovering a tablespace from the backup copy

```
//BI390B  JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*      RECOVER OLAP SAMPLE INDEXES
/*JOBPARM SYSAFF=SC62
//        EXEC DSNUPROC,PARM='DBH1,RECIX'
//STEPLIB  DD  DSN=DB2V61H1.SDSNEXIT,DISP=SHR
//         DD  DSN=DSN610.SDSNLOAD,DISP=SHR
//DSNTRACE DD  SYSOUT=*
//SYSIN    DD  *
   RECOVER INDEX    OLAPR.CUBE1DINDEX1
   RECOVER INDEX    OLAPR.CUBE1DINDEX2
   RECOVER INDEX    OLAPR.CUBE1DINDEX3
   RECOVER INDEX    OLAPR.CUBE1DINDEX4
   RECOVER INDEX    OLAPR.CUBE1DINDEX5
   ............
//*
//* ADD MORE RECOVER INDEX
//*
```

Figure 92. Sample JCL for recovering indexes from the backup copy

6.5.2.2 Using DSN1COPY

In addition to DB2 utilities, you can use the DSN1COPY program to back up and restore DB2 tablespaces and indexes. You should take notes about what has been copied, when the copy has been made, and where the copy has been stored.

```
//BI390A  JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*        STEP 12: LOAD DATA INTO SAMPLE APPLICATION TABLES
/*JOBPARM SYSAFF=SC62
//COPY    EXEC PGM=DSN1COPY
//STEPLIB  DD  DSN=DB2V61H1.SDSNEXIT,DISP=SHR
//         DD  DSN=DSN610.SDSNLOAD,DISP=SHR
//SYSUT1   DD  DSN=DB2V61H1.DSNDBC.OLAPDB.OLAPTS.I0001.A001,DISP=SHR
//SYSUT2   DD  DSN=BI390A.OLAPDB.COPY1,UNIT=SYSDA,
//         DISP=(NEW,CATLG,CATLG),SPACE=(CYL,(150,10)),
//         VOL=SER=TOTDBJ
//SYSPRINT DD  SYSOUT=*
//*
```

Figure 93. Back up a DB2 tablespace data set using DSN1COPY

In Figure 93:

- SYSUT1 defines the data set name to be copied.
- SYSUT2 defines the output data set name.

To restore data that had been copied using DSN1COPY:

1. Set SYSUT1 to the name of the backup file.
2. Set SYSUT2 to the VSAM cluster name of the tablespace or index to be restored.
3. Run DSN1COPY to restore tablespace data or index data.
4. Run the DB2 RECOVER utility with the LOGAPPLY option to apply DB2 log data that is created after DSN1COPY is run, if you want to recover all data to the time just before the database crashed.

For more details, refer to *DB2 UDB for OS/390 Utility Guide and Reference*, SC26-9015.

6.5.2.3 Using DFSMSdss

You can use DFSMSdss to back up tablespace data sets, index data sets, or a whole volume. In this section, we only describe the back up and restore of the VSAM data set because when we use the relational storage manager, all the data and indexes are VSAM data sets. For details on dumping and restoring the HFS files (except data and index), refer to 6.5.1.2, “Using DFSMSdss” on page 159.

You dump one data set and restore it as shown in Figure 94 on page 169 and Figure 95 on page 170.

```
//DFSMSBK JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*
//* BACKUP OLAP DATABASE USING DFSMSdss
//*
/*JOBPARM SYSAFF=SC62
//BACKUP   EXEC  PGM=ADRDSSU
//SYSPRINT DD    SYSOUT=*
//IN       DD    UNIT=3390,VOL=SER=TOTDBK,DISP=SHR
//OUT      DD    DSN=DB2V61H1.OLAPDB.VLSI.SEQ,
//         DISP=(NEW,CATLG,DELETE),SPACE=(CYL,(100,10),RLSE),
//         VOL=SER=TOTDBJ
//SYSIN    DD *
 DUMP DATASET(INCLUDE(DB2V61H1.DSNDBC.OLAPDB.VLSITS.I0001.A001)) -
      COMPRESS TOL(ENQF) -
      LOGINDDNAME(IN) OUTDDNAME(OUT) ALLDATA(*) ALLEXCP
/*
```

Figure 94. Dump one VSAM data set using DFSMSdss

If you want to dump and restore the whole volume, ask your system administrator for assistance.

As we assumed the physical VSAM data set has been deleted, but not uncataloged, we restore the VSAM data set in two steps (see Figure 95 on page 170):

1. Uncatalog the VSAM data set.
2. Restore the data set from the backup file. After restoring the data set, a catalog entry will be made.

If you restore a single data set from a dump of the whole volume, specify the exact file name to be restored (instead of **), for example:

```
include(DB2V61H1.DSNDBC.OLAPDB.VLSITS.I0001.A001)
```

```
//DFSMSREC JOB (999,POK),'RESTORE',CLASS=A,REGION=4M,
//             MSGCLASS=T,TIME=10,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//*************************************************************
//* THIS JOB WILL RESTORE A VSAM CLUSTER FROM A SEQUENTIAL
//* BACKUP IF THE PHYSICAL VSAM CLUSTER HAS BEEN DELETED BUT
//* NOT UNCATALOGED.
//*************************************************************
//* FIRST DELETE THE CATALOG ENTRY
//*************************************************************
//DELETE   EXEC  PGM=IDCAMS
//DD1      DD  DISP=SHR,VOL=SER=SMP7FD,UNIT=3390
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
  DELETE 'DB2V61H1.DSNDBC.OLAPDB.VLSITS.I0001.A001' NOSCRATCH
/*
//*
//*************************************************************
//* RESTORE THE CLUSTER FROM BACKUP
//*************************************************************
//STEP2    EXEC PGM=ADRDSSU,REGION=0M
//SYSPRINT DD   SYSOUT=*
//SEQ1   DD DSN=DB2V61H1.OLAPDB.VLSI.SEQ,DISP=SHR
//DASD1    DD   UNIT=3390,VOL=SER=SMP7FD,DISP=SHR
//SYSIN    DD   *
 RESTORE  INDD(SEQ1) OUTDD(DASD1) TOL(ENQF) -
 DATASET(INCLUDE(**))     -
 REPLACE -
 CANCELERROR
/*
```

Figure 95. Restore VSAM data set using DFSMSdss

6.6 Reorganize DB2 tables

Over time, the tables used by DB2 OLAP Server, especially the fact and key tables, need to be reorganized to reclaim unused space, or to minimize extensions of the tablespaces and indexes after altering PRIQTY and SECQTY size by using a SQL statement. Your database administrator should do this using the appropriate tools. If you are using the DB2 REORG utility, use the following scenario:

1. Determine the tablespaces you want to reorganize by examining the number of extensions of the data set or DB2 catalog tables. (In OS/390, you reorganize the entire tablespace, rather than individual tables.)

In this example, assume that:

The RELCUBEID is 6 and FACTTABLECOUNT is 4; then you will have four fact tables (CUBE6FACT1, CUBE6FACT2, CUBE6FACT3, and CUBE6FACT4).

You have four indexes (CUBE6FINDEX1, CUBE6FINDEX2, CUBE6FINDEX3, and CUBE6FINDEX4).

The key table is either CUBE6KEYA or CUBE6KEYB, depending on what restructures have been performed. The key table index is CUBE6KINDEX.

2. Run the REORG utility on all of the fact tables and the key table. In one job step, you can reorg one tablespace and its indexes only. You need to run the job five times for this example; refer to Figure 96.

```
//OLAPR   JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*
//        EXEC DSNUPROC,PARM='DBH1,REORG'
//*
//SYSUT1   DD  UNIT=SYSDA,SPACE=(CYL,(400,100),,,ROUND)
//SORTLIB  DD  DSN=SYS1.SORTLIB,DISP=SHR
//SORTOUT  DD  UNIT=SYSDA,SPACE=(CYL,(200,100),,,ROUND)
//SORTWK01 DD  UNIT=SYSDA,SPACE=(CYL,(300,100),,,ROUND)
//SORTWK02 DD  UNIT=SYSDA,SPACE=(CYL,(350,100),,,ROUND)
//SORTWK03 DD  UNIT=SYSDA,SPACE=(CYL,(100,100),,,ROUND)
//SORTWK04 DD  UNIT=SYSDA,SPACE=(CYL,(100,100),,,ROUND)
//DSNTRACE DD  SYSOUT=*
//SYSERR   DD  UNIT=SYSDA,SPACE=(4000,(20,20),,,ROUND)
//SYSREC   DD  UNIT=SYSDA,SPACE=(CYL,(400,100),,,ROUND),
//         DSN=UNLOAD.DATA,DISP=(MOD,CATLG,CATLG)
//SYSIN    DD  *
   REORG TABLESPACE DBNAME.CUB6FACT1
//*
```

Figure 96. Sample DB2 REORG job

- Reorg tablespace *dbname*.cub6fact1
- Reorg tablespace *dbname*.cub6fact2
- Reorg tablespace *dbname*.cub6fact3
- Reorg tablespace *dbname*.cub6fact4
- Reorg tablespace *dbname*.cub6keya

Note

In addition, before you load data into a new tablespace, running the REORG utility with the PREFORMAT option is highly recommended to improve load performance.

Chapter 7. OLAP database design on S/390

This chapter provides information about maintaining good performance from a cube design and database allocation perspective.

7.1 Design hints and tips

Some think that they can build a large data warehouse using the OLAP technology and that the whole enterprise data warehouse can be built as an OLAP cube. But this is not so. OLAP cubes (applications) are subject-oriented data marts and *not* enterprise-wide data warehouses.

It is advisable to start very small and grow from there. Once the technology is deployed in an enterprise, it is bound to grow. Our experience shows that companies have deployed many cubes in an enterprise and the user base has grown substantially.

For successful implementation, follow these guidelines:

- Understand your user requirements.
- Involve the users in the process from the beginning.
- Understand the difference between OLAP technology and relational technology.
- Explain the difference between a data warehouse and a subject-oriented data mart.
- Start with a small model.
- Know your batch window and update frequency.
- You may have to consider different techniques for read-only and write back (update) models/applications.

7.2 Comparison of MDB cube and relational cube

Several types of OLAP technology are deployed by various software vendors. There are some major differences between the technology implementations. Select the technology carefully, because once you implement it, you will have to stay within its limitations.

DB2 OLAP Server gives users more flexibility. It gives application designers the best of both worlds. You can choose to implement either a multidimensional storage manager (MDSM) or a relational storage manager

(RSM) method at install time. This is an important difference with other products.

In the MDSM implementation, the OLAP cube is stored in Hyperion's highly optimized proprietary file structure. The data and index components are stored in HFS files, but they can only be accessed by Essbase-ready tools. Hyperion has published their API, so other software vendors can build tools to access the data stored in an Essbase OLAP cube. Currently there are more than 50 tools available to access data stored in Essbase OLAP cubes. There are two files created by MDSM to store data and index components.

The RSM method also offers some benefits. In this implementation, data and indexes are stored in DB2 relational tables. The data is stored in Fact and Dimension tables using star schema format. Since the data is stored in relational tables, users can use Structured Query Language (SQL) tools to access highly optimized OLAP cubes. They can also take advantage of existing Query Management Facility (QMF) queries. Since the data is stored in relational tables, system administrators can take advantage of relational tools for backup and recovery. Although data is stored in relational tables, all relational tables/views are created by DB2 OLAP Server, as the OLAP model is being built. System administrators and database designers do not have to write any data definition language (DDL) statements to create and maintain DB2 objects.

7.3 Data loading

After the application has been designed by a database designer, the next phase is to load the data into an OLAP cube. The data can reside on the same machine or on a machine anywhere in the network. It can also be in various file formats. DB2 OLAP Server can load data from various types of text (flat) files, spreadsheets, or directly from operational data stores on relational platforms or directly from enterprise-wide data warehouses.

Depending on the source data and format, various loading methods are used for data loading. If the source data is in Essbase-ready normal format, it can be loaded directly into an OLAP cube without using any data load rules. If the data is in other formats, then the database designer and system administrator will have to use different load methods to load the data into OLAP cubes. This can be done using the Application Manager interface of DB2 OLAP Server.

Data loading can be accomplished by a variety of methods:

- Free-form
  - Spreadsheet files (.XLS or.WKS)

- Text files

- Lock and Send
- Data load rules object combined with a data source
- DB2 OLAP Server SQL interface

7.3.1 Free-form

Free-form data loading is an easy-to-use process if you keep the following characteristics in mind:

- The data file can be loaded without explicit description of its contents.
- The data must be in natural order, ready to be used by DB2 OLAP Server.
- Any combination of valid dimensions, members, or aliases may be used.
- Data is read according to the member names found in the application outline.

Free-form requirements:

- Each data point must be tagged with a member from each dimension.
- Data must be scanned from the top of the file and from left to right.
- Each field (item) in a record (row) must be separated by spaces, tabs, new lines, or carriage returns.
- A member name or alias must be enclosed in double quotes if it contains blanks, numeric characters, dashes, or unary operators. It must also appear exactly as it does in the outline.
- Members from the same dimension must be in the same column or row.
- Members from different dimensions can be together only on the page header. They cannot be mixed in any other rows or columns.

7.3.2 Lock and send

Lock and send is used to load smaller amounts of data or to make minor changes in data from a spreadsheet (or other similar products).

To lock and send data:

1. Open a spreadsheet program such as Excel or Lotus 1-2-3.
2. Connect.
3. Retrieve a data set (use data that differs from the current database value, that is, data that needs to be updated).

4. Update the values in the spreadsheet.
5. Select **Lock** from the Essbase menu.
6. Select **Send** from the Essbase menu.

This will update the data in the cube. This process is generally used in budgeting and planning types of applications.

7.3.3 Data load with rules file

To load data with rules, follow these six steps:

1. Select **Database--> Load Data** from the Application Manager.
2. Check the **Data files** option button.
3. Find the data file.
4. Check **Use Rules** button.
5. Find the rules file.
6. Check **Load data, Interactive**. Click **OK**.

When data load is done, the Data Load Completed dialog box appears and displays the load status; see 5.4.1, “Preparing the Sample application” on page 130.

The proper use of the data load method is an important part of application design. Designers must keep in mind the data load frequency when building an application model. Various techniques can be implemented, depending on whether data is being refreshed (replaced) or modified (incremental updates). Another important consideration is the batch window available to load the data.

If you are using an MDSM implementation, the data is loaded in the essnnnnn.pag file. As data is being loaded, the index is also being built. The index information is stored in the essnnnnn.ind file. Both files, by default, reside in the HFS file structure in /essbase-home/app/application-name/database name/.

If you are using an RSM implementation, the data is loaded in various relational tables. The *fact and dimension* tables are updated as data is being loaded into OLAP cubes. DB2 system catalogs are updated automatically by DB2 OLAP Server, as it builds new DB2 objects. The information about the application, its cube names and related DB2 table and view names, are stored in a DB2 table named supervisor-ID.CUBECATALOG.

7.3.4 DB2 OLAP Server SQL interface

The DB2 OLAP Server SQL interface is used to load data directly from the relational database. This method can be very useful when you have a large relational data source, as you do not have to extract the data to a different source for loading.

To load data with the SQL interface, follow these steps:

1. Select **Database--> Load Data** from the Application Manager.
2. Check the **SQL type** button.
3. Type the SQL user ID and SQL password.
4. Check **Use Rules**.
5. Find the rules file.
6. Select the relational database and table.
7. Check **Load Data, Interactive**. Click **OK**.

When the data load is done, the Data Load Completed dialog box appears and displays the load status.

Table 7 summarizes various load process for different input sources.

Table 7. Load process and the type of input sources

Load process	Source file format
Free Forms	Text file, Spreadsheet
Lock and Send	Spreadsheet
Rules File	When you want to load selected columns or need to perform data manipulation Text file, Spreadsheet, SQL table
SQL Interface	SQL table

7.4 Incremental data loading

Although it is easy to reload the entire database, it is not always possible as this can take very long time. DB2 OLAP Server allows you to add/append data to existing application/data. The incremental load can be achieved as follows:

1. Select **Database--> Load data** from Application Manager.

2. Select the **Rules editor** button in Application Manager. This is the first button on the right, that will give a list of rule files associated with the application and database.
3. Open the Rules file you are using to load the data.
4. Click the **Global Data Load Attributes** button, or select **Options--> Dataload Settings** to open the Data Load Settings dialog box.
5. Click the **Data Values** tab: see Figure 97.
6. In the Data Values tab, select **Add to existing values**, then click **OK**.

Figure 97 shows this process.

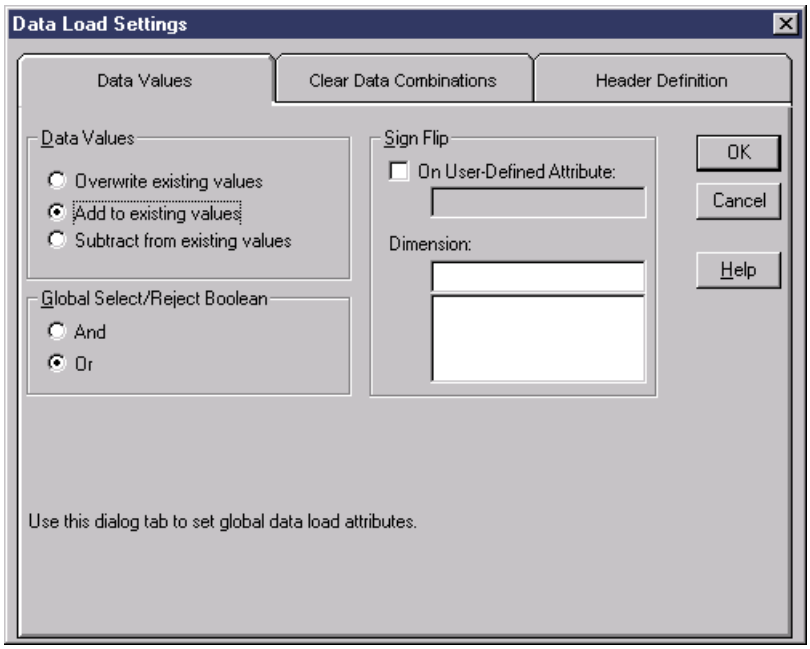


Figure 97. Data load setting for incremental loading

Note that once the data is loaded, you must perform calculations to reflect the new data.

7.5 Calculation

Once the data is loaded into an OLAP cube, the next step is to perform calculations. This step must be performed before users can start using the OLAP cube to analyze the information. In this step, all pre-calculations are done and the cube will contain all possible information. This allows users

"Speed of Thought" response time. The data and index files grow substantially during this phase. This step can take much longer than the data loading step.

There are three basic methods for calculating a database:

- Outline consolidation
- Custom calculation
- Intelligent calculation

7.5.1 Outline consolidation

The simplest method of calculation is based on the database outline. This method is referred to as the default calculation. In this method, all pre-calculations are done and the cube will contain all possible information. This delivers a "Speed of Thought" response time. The data and index files are enlarged substantially during this phase. This option can take much longer than the data loading step.

In this method, database calculation is based on the default consolidations implied in the database outline. Member formulas are calculated as they are defined in the outline.

Outline consolidation exhibits the following characteristics:

- Calculations that aggregate and roll-up data based on the hierarchical relationships of the outline
- Relationships that include computed members defined by formulas
- The use of unary operators in the outline (+,-,%,/. and ~)

For more information, see *Essbase Database Administrator's Guide,* SC26-9238.

Calculation order - all dimensions

When a default calculation occurs, DB2 OLAP Server calculates all dense dimensions within each data block before moving on to subsequent blocks.

- Accounts dimension is calculated first.
- Time dimension is calculated second.
- Remaining dense dimensions are calculated after Accounts and Time.
- Remaining sparse dimensions are calculated in the order in which they appear in the outline.

7.5.1.1 Dynamic calculations

Dynamic Calc members are defined by tags set within Outline Editor.

Dynamic Calc members do not require pre-calculation during a normal batch calculation process. Instead, Dynamic Calc members are calculated in response to specific retrieval requests from Spreadsheet Add-in, from Report Writer, or from distributed online analytical processing (OLAP) operations.

Dynamic calculations are performed upon retrieval of data. Since the data is not pre-calculated in this method, it takes a little longer to retrieve the data, but it has several advantages:

- Smaller batch window

  Since the data is not pre-calculated the total time required to perform load and calc functions is greatly reduced. This helps a great deal when the batch window is very small.

- Reduce disk storage

  Since the data is not pre-calculated, the upper-level blocks are not created for members tagged as Dynamic Calc. These blocks are created only at retrieval time, thus reducing disk space utilization.

You should consider this option if you have any of these conditions:

- Small batch window
- Upper-level dense members or formulas
- Sparse members with small fan-out

Comparing dynamic calculation members

There are multiple options for dynamic calculations. You can choose different options, depending on the scenarios. Dynamic Calc members can be store or non-store:

- Dynamic Calc and store members
  - Upper-level sparse member on remote server
  - Upper-level sparse member with complex formula
- Dynamic Calc members (non-store)
  - Dense member
  - Sparse member with small fan-out

Dynamic calculation and store
When using this option, the data blocks are created only upon retrieval. The data blocks will be stored after the first request. This will reduce the access time for all subsequent requests as the data blocks will be stored after the first access. You should consider this option in the following scenarios:

- Small batch window
- Upper-level sparse members with complex formulas

Dynamic Calc members reduce batch calculation time and lower disk usage. Thus, dynamic calculation benefits customers who need small calculation windows or have limited storage capacity.

Because dynamic calculations occur only during retrieval requests, retrieval time increases in proportion to the number of dynamic calculations that need to be computed.

Dynamic Calc (non-store)
Dynamic Calc (non-store) members discard calculated values after completing a retrieval request. Choose Dynamic Calc (non-store) members for data values that require iterative calculations—budgeting, planning, forecasting models, and so forth.

Both dense and sparse dimension members can be tagged Dynamic Calc:

- Dynamic Calc members can be consolidated members or members with a formula
- Parents of Dynamic Calc members can be either real (not tagged) or Dynamic Calc.
- Some members cannot be tagged as Dynamic Calc, specifically:
  - Level zero (0) members without formulas
  - Label-only members
  - Shared members

In a single parent-to-child case, if the child member is dynamically calculated, then the parent member must also be dynamically calculated.

7.5.2 Custom calculation

Custom calculations differ from those defined in the outline. Generally custom Calc scripts are written to achieve this. A calc script can override either the order of the default calculation or the member formulas in the outline.

Calc scripts

Calc scripts define calculations that differ from those defined in the database outline. By separating calculation logic from the outline, you can quickly design and run custom database calculations.

Calc scripts can be used to:

- Calculate a subset of the database
- Perform currency conversions
- Define a calculation order that differs from the default
- Calculate member formulas that differ from those in the outline
- Perform multiple-pass calculations for allocations, goal setting, or other complex requirements
- Clear or copy data from specific members
- Define and use temporary variables

7.5.3 Intelligent calculation

Intelligent calculation achieves the following:

- Calculates only those blocks and the ancestors of those blocks that have changed since the last calc
- Proves most effective for sparse incremental updates

The calculator has built-in intelligence to recalculate only those blocks that have been modified since the last calculation. Often you update data values in the database through an incremental data load or by modifying spreadsheet figures with a lock and send. When you update input values, you often want to recalculate the database. In this case, DB2 OLAP Server employs a special method for recalculation called intelligent calculation.

Intelligent calculation method

The intelligent calculation method works as follows. It:

- Looks at whether data blocks in the index are dirty
- Upon recalc, performs calcs only on blocks marked dirty
- Keeps flags in the index, not in the block
- Scans the index to determine which blocks are to be calc'ed
- Clears the dirty status after calculation and marks blocks clean
- Reduces calculation time for incremental updates

7.6 Optimization tuning tips

There are four areas of optimization in DB2 OLAP Server. Use the appropriate method for your environment.

7.6.1 Outline optimization

Once your outline is created, you need to take a look at how the dimensions and members are ordered. Your data consolidates based on the order of dimensions and members within your outline. Within the database outline, you can apply tags to dimensions and members to affect how DB2 OLAP Server treats your data. In the database outline, dimensions should be placed in the following order:

- Dense dimensions: From small to large
- Sparse dimensions: From small to large

7.6.2 Calculation optimization

You can apply calculation formulas to your outline also. There are various calculation optimization techniques, including the following:

- Dynamic calculation
- Dense before sparse
- Repeating commands
- IF versus FIX (Each FIX will cause one pass through the database)
  - IF for dense dimensions
  - FIX for sparse dimensions (use FIX instead of multiple IF commands on sparse dimensions)
- Additional calculation settings:
  - SET CALCHASHTABLE command

    Use this command for any database that has more than 5000 children.
  - SET FRMLBOTTOMUP command

    It optimizes the calculation of formulas on sparse dimensions in large database outlines.

7.6.3 Dataload optimization

The format and layout of input data can also have a big impact on the total time it takes to load the data. There are several techniques that can be applied to speed up the load process.

- Sparse dimension ordering

  This technique implies that the input data should be sorted in the descending sparse dimensions order (large sparse—small sparse).

- Dense dimension ordering

  This technique implies that the input data should be sorted in the descending dense dimensions order (large dense—small dense).

- Essbase ‘export’ command

  When you use the ‘export’ command either from ESSCMD or Application Manager, the Hyperion engine will write the data to an external file in block format (that is, one entire block at a time). This file can be used for subsequent load/restore. Since the data is exported in blocks, it reduces the I/O time to load the data.

- Sort sparse dimensions

  Sort the input file in sparse dimensions order (all sparse dimensions followed up by all dense dimensions).

- Ordering of dense dimensions

  Place dense dimensions closest to data. One dimension should go across data—this will decrease data load times by 20 to 30 percent on average.

7.6.4 Database settings

This is performed from **Database--> Settings** in the Application Manager.

- Index cache: large enough to keep 100 percent of blocks in memory.
- Data cache: the target should be 25 percent to 35 percent of data blocks in memory.
- Allocate more memory for the index cache than the data cache.

7.7 Performance improvement techniques

Several techniques can be applied for performance improvement:

- Minimize the number of passes through data blocks
- Place more data into the block

  This will allow the calculator to calculate more data with less I/O. It will also reduce the size of the index, providing the benefit of faster restructure times.

Considerations:

- Block size between 10 KB—50 KB (never more than 200 KB)
- Do not *Daisy Pick* dimensions

Choosing dense/sparse dimensions
First you must determine if you have a requirement for sparse matrix management.

Then, choose between the techniques using one of these basic approaches:

- Intuitively—by studying the data
- Mechanically—by using a testing protocol to iterate through choices and measuring the results

A testing protocol
The job of the OLAP designer is to create the smallest number of the densest blocks possible. This implies that the dense/sparse configuration which has the maximum ratio of block density divided by the number of blocks will have the fastest calculations.

A critical ratio
This can be expressed as the fraction:

```
Block Density/Number of blocks
```

To determine this value:

- Get the block density measure from dbstats.
- Get the number of blocks from dbstats after a full calc of the database.

Performance improvement summary
Use the following steps to determine performance improvements:

1. Choose a dense/sparse configuration.
2. Compute the block size.
3. Put Dynamic Calc on every dense member that is a parent or has a formula.
4. Clear the database.
5. Load the base blocks. Run a Set msg only calc.
6. Compute the ratio: block density/number of blocks.
7. Capture the results.

7.7.1 RSM implementation

The anchor dimension is always one of the dense dimensions. By default, DB2 OLAP Server will try to use the *measure* dimension, which you mark with an "*accounts"* tag in Application Manager. DB2 tables do have a limit on number of columns. If the measure dimension has too many members to allow us to create one column per member, then DB2 OLAP Server looks for a dense dimension that will fit. You can also specify the anchor dimension in Application Manager.

If it is taking a long time to open an Essbase application or database using the Application Manager, or to select it using ESSCMD, check the number of extents for DB2 tablespace you are using (as specified by the TABLESPACE parameter in your rsm.cfg file). If this number is high (that is, over 50), performance could suffer. Reorganizing the tablespace to reduce the number of extents might improve performance.

7.8 How to optimize data placement

It is ideal to allocate OLAP data and indexes on separate volumes for each DB2 OLAP Server application. In this section we introduce how to place the OLAP tables and indexes over several DASD volumes to minimize I/O time.

7.8.1 Using MDSM

Create multiple HFS files on the different volumes and mount the files. Before loading a database, you can use ESSCMD or Application Manager for this. The following example is done using Application Manager:

1. Click **Database--> Setting** in main tool bar.
2. Go to **Storage**.
3. Put the *directory name* you want to store data into in *Disk Volume.*
4. Select the *File type* you want to store: Data, index, or both.
5. After data loading, the data and index will be stored where you have specified.

Figure 98 on page 187 shows an example of this. In our example,

- Data from the Demo database is stored in the /u/data/Demo directory.
- Index from the Demo database is stored in the /u/index/Demo directory.
- These two directories are pointing to different HFS file systems.

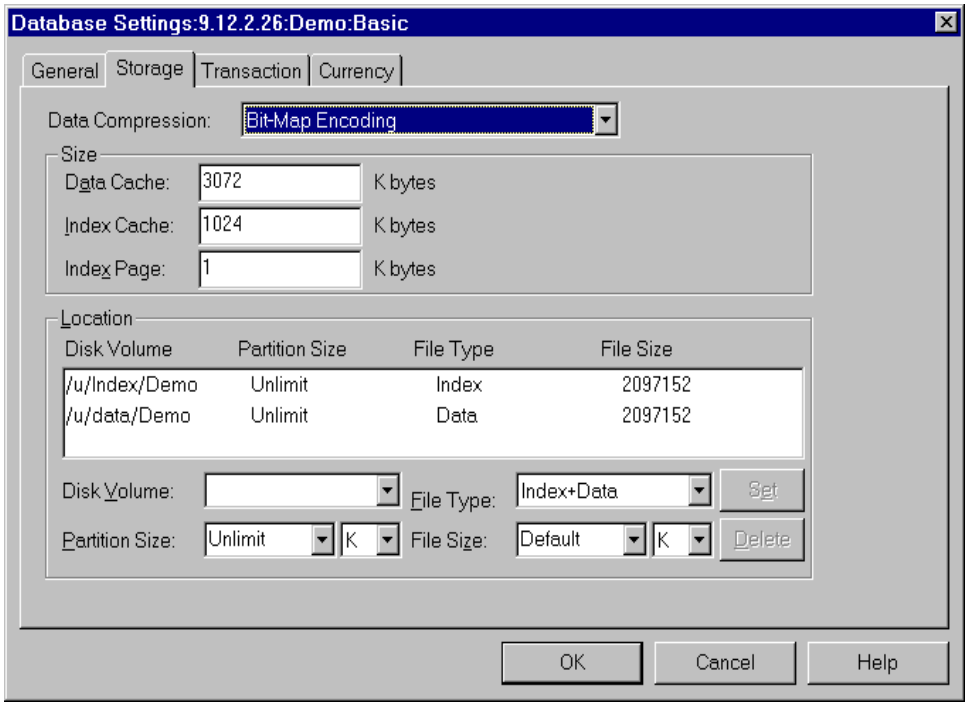


Figure 98. Storing data and index in different HFS systems

7.8.2 Using RSM

When designing your storage architecture for DB2 OLAP Server on OS/390 for RSM implementation, follow these guidelines:

- Use different storage groups on different volumes.
- Put each fact table in a separate tablespace in separate storage groups.
- Put each fact table index in a separate storage group.
- Put the key and dimension tables in a different tablespace than the key and dimension table indexes.
- For administrative tables, use 32 KB tablespaces.
- Spread tables and indexes on different DB2 bufferpools.
- Run the REORG utility with PREFORMAT option before loading new tablespaces.
- Run the RUNSTATS utility after load and before calculation.

Note

DB2 OLAP Server for OS/390 does not support DB2 partitioned tablespace. The partitioning parameter in rsm.cfg also is not relevant to DB2 OLAP Server for OS/390.

7.8.2.1 Allocate a separate DB2 database for the application

At installation time, you created one database and two tablespaces for the sample applications. When you want to use different DB2 databases or tablespaces for a new application, you should use the following steps:

1. Create several DB2 storage groups with different physical DASD volumes.
2. Create a new DB2 database, assigning a DB2 storage group to the database, which will be used for indexes as a default storage group.
3. Create a new tablespace, using a different storage group that will be used for tables.
4. Change the rsm.cfg file to reflect the new database and tablespace names; see Figure 99.

```
[RSM]
RDB_NAME=DBH1
TRACELEVEL=0
ISOLATION=CS
TABLESPACE=IN OLAPDB.OLAPTS
ADMINSPACE=IN OLAPDB.ADMINTS
[VLSI]
RDB_NAME=DBH1
TABLESPACE= IN RSMDB.VLSITS
<Vlsi>
TABLESPACE= IN RSMDB.VLSITS1
KINDEX= USING STOGROUP STOVLSI1 PRIQTY 2000 SECQTY 1440
FINDEX= USING STOGROUP STOVLSI2 PRIQTY 2000 SECQTY 1440
<Vlsisale>
KINDEX= USING STOGROUP STOVLSI3 PRIQTY 4000 SECQTY 1440
FINDEX= USING STOGROUP STOVLSI4 PRIQTY 4000 SECQTY 1440
```

Figure 99. Sample rsm.cfg to distribute data of an OLAP application

Figure 99 notes:

- [RSM] describes the general and global definition of the OLAP applications and the databases.
- [VLSI] begins the application section of the new application VLSI.
- Generally, all tables of the VLSI application are created in the database RSMDB and tablespace VLSITS when a separate database is not specified in the database section.
- The tables of the Vlsi database of the VLSI application are created in RSMDB.VLSITS1; see <Vlsi> database section. In contrast, the tables of the Vlsisale database are created in RSMDB.VLSITS.
- The indexes of the key table and fact tables are created using different DB2 storage groups with a separate primary allocation size of DASD (KINDEX, FINDEX). This clause is used to create an index statement in DB2; the unit size of PRIQTY is KB.

5. Create a new application and a new database from Application Manager or ESSCMD.
6. Load data into the new database. After data loading, your table will be stored in different DASD volumes than where the indexes are stored.

Note

Whenever you modify an application or database section in rsm.cfg, you must stop and start DB2 OLAP Server to make the new definition effective. Before saving a database outline, you should modify the application or database section in the rsm.cfg file.

7.8.2.2 Allocate fact tables into the separate DB2 tablespaces

In RSM, you can not use DB2 partitioned tablespaces to distribute data.

DB2 OLAP Server always splits the fact table into four different (physical) tables, and the names are CUBEnFACT1, CUBEnFACT2, CUBEnFACT3, and CUBEnFACT4, where n is the cube number. You can put these four fact tables into four different tablespaces by specifying the FACTS parameter in the rsm.cfg configuration file using the following steps:

1. Create several DB2 storage groups.
2. Create four tablespaces in which the four fact tables will reside. Use the same tablespace-naming convention with a different suffix (1 to 4). The

tablespace names should be matched with the FACTS parameter in the rsm.cfg configuration file.

3. The TABLESPACE parameter specifies the tablespace to use for the key table and dimension tables.
4. The FACTS parameter specifies the tablespaces for the fact tables and indexes that DB2 OLAP Server manages. You must use a *question mark* in place of the table number in the parameter. DB2 OLAP Server generates a number from 1 to 4 to replace the question mark.

```
[TESTFACT]
RDB_NAME=DBH1
TABLESPACE= IN OLAPDB.TESTTS
<Vlsif>
FACTS= IN OLAPDB.FACT?
KEYSPACE= IN OLAPDB.KEYTS
KINDEX= USING STOGROUP STOVLSI5 PRIQTY 6000 SECQTY 1440
FINDEX= USING STOGROUP STOINDX? PRIQTY 3000 SECQTY 1440
```

Figure 100. Sample rsm.cfg to distribute four FACT tables

5. The FINDEX parameter specifies the storage group for the indexes of fact tables that DB2 OLAP Server manages. For details, see 7.8.2.4, “Allocate large DASD space for indexes” on page 191.
6. The KINDEX parameter specifies the storage group for the indexes of key tables. For details, see 7.8.2.4, “Allocate large DASD space for indexes” on page 191.

7.8.2.3 Allocate separate tablespace for a key table

It is highly recommended to allocate a separate tablespace for a key table which is the second largest table in a relational cube. The KEYSPACE parameter provides a string that DB2 OLAP Server appends to the CREATE TABLE statement when you create a key table. (In fact, when you save a database outline, all tables are created.)

You can improve DB2 OLAP Server performance by placing this tablespace on a fast storage device.

As shown in Figure 100, the key table is created in the OLAP database and KEYTS tablespace, while the fact table is created in FACT1—4 tablespaces and other tables are in the TESTTS tablespaces. If you do not specify this parameter, the key table is stored in the TESTTS tablespace specified in TABLESPACE parameter.

7.8.2.4 Allocate large DASD space for indexes

Most indexes for the database (or cube) are stored in the DB2 default storage group of the database. For example, all indexes for dimension tables and alias tables are stored there.

When you are loading or calculating with a large volume of data, you need more DASD space for the fact table index and key table index, and you may want to store these indexes in a separate DB2 storage group. You can use the FINDEX, KINDEX parameter of rsm.cfg to achieve this.

You can specify separate DB2 storage groups for the indexes of fact tables and key table that DB2 OLAP Server manages. DB2 OLAP Server appends the clause you enter here to a DB2 CREATE INDEX command. For example:

```
FINDEX=USING STOGROUP OLAPSG? PRIQTY 6000 SECQTY 3000
BUFFERPOOL BP2

KINDEX=USING STOGROUP OLAPSGK PRIQTY 5000 SECQTY 2000
BUFFERPOOL BP2
```

Where ? can be the number of the fact table. You must use a question mark.

The DB2 OLAP Server supervisor ID needs DB2 privilege to allocate space for indexes into DB2 storage groups. So the DB2 system administrator should enter the following:

```
GRANT USE OF STOGROUP stogroup-name to supervisor ID;
```

By default, the PRIQTY for an index is one track, which is rather small for production data. Most of the time you need to allocate a size large enough to handle all index records. Use a large number in PRIQTY and SECQTY. The unit size is KB.

For the index of the key table, you can specify a different storage group, and primary and secondary quantity.

7.9 How to improve load and calculation processes

There are several options to improve elapsed time for loading data into the database. This section describes them.

Managing cache size

Before loading data into the database, you can adjust cache size for data and indexes, which improves your load and calculation performance.

Select **Database--> Setting--> Storage** to adjust values. As shown in the size section of Figure 98 on page 187:

- The default cache size for data is 3072 KB.
- The default cache size for index is 1024 KB.

These values are not enough for large database. We usually size memory based on cache requirements for calculation. The rule of thumb is you get the best results with enough memory to cache the entire cube index, and three times that amount for the data cache.

After data loading you can find statistical information for the database, such as data block size, number of blocks, and so on, from Application Manager.

Managing commit block number

The commit block value in database setting should be modified. In most cases, the default of 3000 is rather small. (10000 blocks seems better.)

Chapter 8. Capacity planning

When capacity planning for OLAP, we need to estimate the machine requirements for:

- CPU
- Memory
- DASD

Our goal is a machine specification that has sufficient power and storage to build cubes and deploy them to analysts.

Unlike many other server technologies where we focus mainly on the base data or the number of users we need to support, the key consideration for OLAP is the time and space required to calculate and store the OLAP cubes. For an OLAP cube to be useful, it must be regularly updated with new data, and then calculated, before being made available to the analysts. As data flows into the data warehouse at the end of each day, week, or month, we refresh the cube with a new set of data. This operation must be accomplished in a certain time. Typically, we know how long the batch window is, and how much of that time we can allocate to building OLAP cubes, so we can plan to meet these requirements.

8.1 Inflation

DB2 OLAP Server is a multidimensional OLAP technology. This class of OLAP tools offers incredibly fast and consistent query response times. To make query response times consistently fast, we build an OLAP hypercube, so that we minimize the work the server needs to carry out in order to supply data to the analysts.

Unlike the base tables in a data warehouse, OLAP hypercubes contain aggregates. It is these aggregates that result in cubes exploding to many times the size of the input data.

Consider a simple two-dimensional model in which we load monthly data for sales, cost of goods, and expenses as shown in Table 8 on page 194. If we lay out these members as a grid, we have a 3 X 12 grid of 36 cells.

Table 8. Example of two-dimensional model

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Sales	100	120	170	200	200	200	190	180	185	185	220	250
Cost of goods	75	80	110	140	140	140	120	120	125	130	160	190
Expenses	10	10	10	15	15	15	15	15	15	15	20	20

To add value to this simple model, we add some simple aggregates. We will rollup the months into quarters, and the quarters into years. We will calculate margin as sales—cost of goods, and profit as margin—expenses, then we will compute profit and margin as a percentage of sales. Although our model is simple, we have added significant value. Analysts can now track margin and profit, and some key percentages, and we can see how the business is performing at the yearly, quarterly, and monthly levels.

We have increased the number of members in the time dimension by 5 (that is, one member for each quarter, and one for the full year). This is approximately a 42 percent increase. We have increased the number of members in the measures dimension by 4, which is approximately a 133 percent increase. But if we look at what has happened to our simple grid, it is now 17 X 7, that's 119 cells; so our simple cube has inflated from 36 cells to 119 cells, that's approximately 300 percent.

Table 9. Example model with aggregates added

	Jan	Feb	Mar	Q1	Apr	May	Jun	Q2	Jul	Aug	Sep	Q3	Oct	Nov	Dec	Q4	Yr
Sales	I	I	I	G	I	I	I	G	I	I	I	G	I	I	I	G	G
Cost of goods	I	I	I	G	I	I	I	G	I	I	I	G	I	I	I	G	G
Margin	G	G	G	D	G	G	G	D	G	G	G	D	G	G	G	D	D
Expenses	I	I	I	G	I	I	I	G	I	I	I	G	I	I	I	G	G
Profit	G	G	G	D	G	G	G	D	G	G	G	D	G	G	G	D	D
Profit %	G	G	G	D	G	G	G	D	G	G	G	D	G	G	G	D	D
Margin %	G	G	G	D	G	G	G	D	G	G	G	D	G	G	G	D	D

Note the following in Table 9:

- I shows input cells

- D shows derived x derived
- G shows that we have generated input x derived.

In Table 9 on page 194, we have 36 cells (I) that we loaded with input data values. We calculated 20 cells (D) based on derived members in both dimensions, such as profit for Q1. We have also added 63 cells (G), where we cross a derived member with a base member, such as sales for Q1 and profit for January. What we often fail to consider is the effect of the additional 63 cells: These are the cells that inflate OLAP cubes. When you consider this effect across 5, 6, or 7 dimensions, you begin to understand why our OLAP cubes are much larger than the input data. This effect is known as *inflation*. We suggest that inflation can be calculated by the following methods:

1. Calculate the ratio of total members to input members for each dimension of the cube.
2. Average the ratio.
3. Raise the average ratio to the power of the number of dimensions.

In the example we used above, we find that the averaged ratio is 1.85. If we take this ratio and apply it to a 5-dimensional model, we get an inflation factor of 21, and for 8 dimensions we get an inflation factor of 137. This means that for an 8 dimensional model with an inflation factor of 1.85, every megabyte of data we load will generate 136 megabytes of data when we calculate the cube.

Although the inflation factor cannot be used to estimate cube size, it can be used to quickly test your design, as soon as you understand the number of dimensions and the number of members in each dimension.

We calculated the inflation factor for a number of cubes, and found an average inflation factor of 3.3, the highest being 4.6.

8.2 Sparsity

Our inflation factor can give us an early indication of a bad design, but it does not take into account the sparsity of the data. Today's OLAP cubes are very sparse. As we build larger models that hold more detailed data, they will become even sparser.

Why is enterprise data so sparse? To answer this question, let’s consider a simple example. Widgets Inc., sells Real Widgets, and for large retailers it creates branded versions, labeled for the retailer. If we look at the sales

figures for November 1999, then we might see something like the figure shown Table 10:

Table 10. Example widgets model that exhibits sparsity

	Real Widgets	**Acme Branded**	**Gonzales Branded**	**Robinson Branded**
Acme retail		200,000		
Gonzales retail			1,000,000	
Robinson retail	250,000			250,000

Acme, Gonzales, and Robinson retail are not purchasing widgets branded for their competitors. Some retailers may stock both generic Real Widgets, and their own brand. Many smaller retailers will only stock Real Widgets. The matrix across many products and customers will be very sparse.

Now consider a real example such as a bookstore. They may stock many thousands of titles, but each customer will only purchase a very small number of titles each year. Even if we rollup titles into categories before loading the OLAP cube, we will still have a very sparse matrix. Think of the categories of books available at your local bookstore, and imagine the matrix for *your* purchases in the last year. This is sparsity.

Sparsity makes an enormous difference to the efficiency with which DB2 OLAP Server can store the data. Using a patented algorithm, DB2 OLAP Server views the dimensions of a model as either sparse or dense. For combinations of members in the sparse dimensions for which at least one data point exists in the combinations of members in the dense dimensions, DB2 OLAP Server stores an index entry, and a block to hold the data values. In our widgets example, Product and Customer are our sparse dimensions, so DB2 OLAP Server will create index entries and blocks to store the sales figures for the combinations:

- Acme Retail -> Acme Branded
- Gonzales Retail -> Gonzales Branded
- Robinson Retail -> Real Widgets
- Robinson Retail ->Robinson Branded

Although there are 12 possible combinations of Product and Customer, we only manage the four combinations for which we have data, so the percentage of potential combinations we store is 33 percent. This ratio, which is usually referred to as the percentage of maximum blocks existing, is key to

understanding how many data blocks DB2 OLAP Server must store, how large the final cube will be, and how long it will take to calculate.

We recently carried out some tests using small data sets with varying data densities. The results are shown in the charts in Figure 101. With the percentage of maximum blocks varying between 0.037 percent and 3.455 percent, we saw storage requirements increase 60 times, and calculation times increase by a factor of 162.

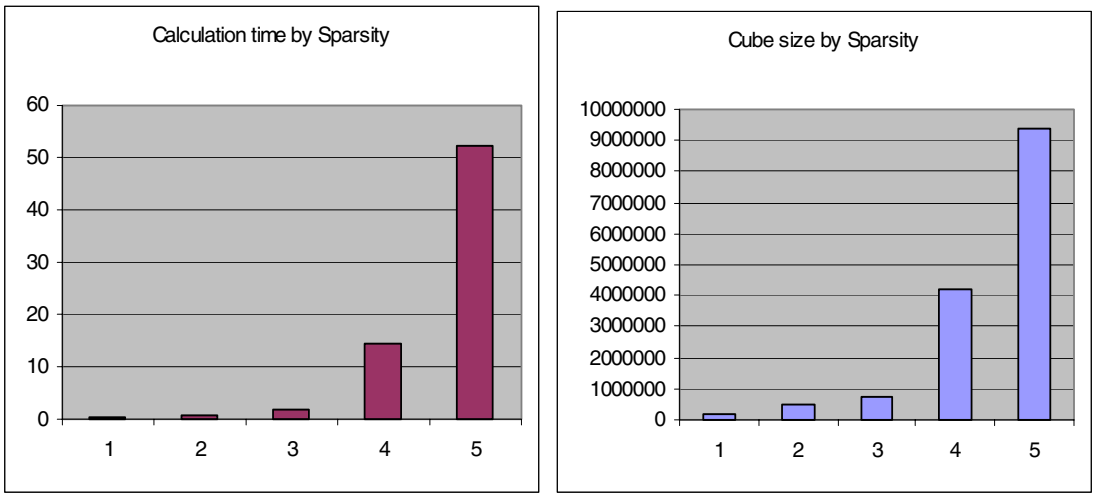


Figure 101. Cube storage requirement and calculation time versus sparsity

Unfortunately, estimating sparsity is very difficult. As E. F. Codd states in his paper *Providing OLAP (On-Line Analytical Processing) to User-Analysts: An IT Mandate,* "...if the enterprise data is sparse, and has certain distribution characteristics, the resulting physical schema might be one hundred times the size of the enterprise data input. But, given the same size data set, and the same degree of sparseness, but with different data distribution, the size of the resulting physical schema might be only two and one half times the size of the enterprise data input, as in the case of the perfectly dense example. Or we could experience anything in between these two extremes. Eyeballing the data in order to form an educated guess is as hopeless as is using conventional statistical analysis tools to obtain cross-tabs of the data."

For many of today's OLAP models, the percentage of maximum blocks existing will be as low as 0.2 percent.

8.3 Block density

The final factor we need to consider when estimating model size are the characteristics of the dense blocks that DB2 OLAP Server stores. As we saw in 8.2, “Sparsity” on page 195, we store one block for each combination of sparse members where at least one value exists in the dense member combinations. In our simple example, we had just one measure to store from our single dense dimension. Real models will usually have a number of measures, and more than one dense dimension. Time and Scenario are frequently the other dense dimensions.

When deciding which dimensions to tag as dense, we must consider the size of the resulting block. We can calculate block size by multiplying together the number of members in each dense dimension, and then multiplying by 8. This gives us block size in bytes. We should aim for a block size that is between 8 KB and 100 KB.

When DB2 OLAP Server stores the blocks, it carries out compression. Although these blocks are dense, relative to the other combinations of dimensions, data densities in these blocks are still low, often between 5 percent and 10 percent. Compression removes the empty cells, resulting in much less I/O and high storage efficiencies. The ratio of compressed block size to full block size is known as the *compression ratio*. Ratios between 0.4 and 0.8 are common.

8.4 Gathering Information

In order to begin the estimating process, we must first understand how many cubes we plan to build. Each cube should be targeted at a business process or a business question.

For each of the cubes we must understand the number of dimensions and the number of members in each dimension. If we can determine the ratio of input members to total members in each dimension, then we can check the inflation factor.

Next, we must decide which dimensions are dense and which are sparse. Using the member counts for the dense dimensions, we can check that our block size is in-range.

Now we must concern ourselves with the characteristics of the data. How sparse is the index, and what storage efficiency can we expect for the dense

blocks? This is by far the hardest part of the process, and the resulting numbers have enormous impact on the estimates.

Because these factors are very difficult to estimate, and yet critical to the process, we strongly suggest that the values should be determined by working with the data, either by building a prototype or by conducting a proof of concept (POC).

8.4.1 Proof of concept

Using a proof of concept method is by far the most reliable method of gathering the information we need. Typically, this method will result in both a good design and in estimates that are accurate to within 20 percent, even though we do not build the complete model.

In order to gather the key metrics, percentage of maximum blocks existing, and compression ratio, we will build two slices through the cube. In the first case, we subset the dense dimensions and measure the percentage of maximum blocks existing. This gives us the ratio across the sparse dimensions. In the second case, we subset the sparse dimensions and measure how efficiently we store the blocks. This is the compression ratio across the dense dimensions.

We have built all dimensions, and improved our understanding of the model, but because we load and calculate subsets, we do not have to wait for the entire model to build. This method gives us accurate metrics in a fraction of the time it would take to build the complete model, and it is far more reliable than sampling.

We applied this technique to the model illustrated in Table 11:

Table 11. Example proof of concept model

Dimension	Members	Stored Members
Time	86	32
Measures	15	6
Scenario	16	2
Channel	6	5
Product	154	132
Customer	1001	1000

For our dense slice, we loaded and calculated one fiscal quarter, thus reducing the number of stored members in the time dimension from 32 to 4. When the calculation is complete, the percentage of maximum blocks existing can be found on the statistics tab of the database information dialog in Application Manager. With this subset, the percentage of maximum blocks was reported as 3.46 percent, the same as the full model.

Next, we built a sparse slice through the product dimension. To do this, we took one family of five products, thus reducing the number of stored members in the product dimension from 132 to 5. When the calculation is complete, the compression ratio can be found on the statistics tab of the database information dialog in Application Manager. When we loaded and calculated this subset, we got a compression ratio of 0.11. The full model has a compression ratio of 0.10.

As long as you choose the dense and sparse slices carefully, this technique will yield excellent results. Note that you are dealing with data that is extremely seasonal, you may need to build subsets for off-season and peak periods, and factor the different metrics into your estimate.

8.4.2 Estimating from a warehouse

If you are unable to build a prototype or conduct a proof of concept, then some knowledge of the data is required. Rather than simply guessing, we suggest you work with your information technology (IT) organization to build SQL queries to get an indication of density.

If you are working with a dimensional data warehouse, then it will be very easy to query the dimension tables to determine how many members each dimension will have.

Next, you must decide which dimensions are dense and which are sparse.

If we multiply the member counts for each sparse dimension, we know the maximum number of potential blocks. To find out how many of these will exist in the OLAP input data, run an SQL query against the fact table. Each unique combination of members in the sparse dimensions that exist in the fact table is an input block. Of course, we will have to carry out some grouping in order to work with the summary-level data we will load into the OLAP server, rather than the atomic data in the fact table. The ratio of actual input blocks to the maximum number of input blocks is the sparse ratio we need.

Similarly, we can use the member counts in the dense dimensions to estimate maximum block size. If we run a series of SQL queries which determine how

many NULL values each input block will contain, we can use the ratio of NULLs to values as an estimate of compression ratio.

8.4.3 Estimating without data

If you cannot conduct a proof of concept, or work with IT to investigate the data, you have no choice except to make some educated guesses. Rather than simply pulling a number out of the air, we suggest you conduct an exercise in which you first define your dimensions and hierarchies, then estimate your member counts, and then nominate dense and sparse dimensions.

Make a grid, like the one in Table 10, using the members of a reasonable subset of two of your sparse dimensions. We might expand table 10 to cover all products, and a subset of all customers, such as those in one region. We complete the grid by marking each intersection where we expect to find data. The percentage of marked intersections gives us an estimate of sparsity for these two dimensions.

Now we need to consider how this estimate will change when we add the additional sparse dimensions in the model. In our example, Channel might be the remaining sparse dimension. The estimate we made was for all channels, so we use this value as a base, and we estimate what percentage of cells will be populated for each channel. If we had two sparse dimensions, we could construct a second grid such as Channel by Package, and estimate the percentage of populated cells at each intersection in this grid.

As we factor in each sparse dimension, we calculate a new estimate for percent of maximum blocks existing, and we find that the estimate gets lower. When we have considered all the sparse dimensions, we have a final estimate.

To estimate block density, we repeat this exercise for the dense dimensions, and use the ratio of populated cells to empty cells as an estimate of the compression ratio.

8.5 Using the information

Now that we have gathered the required information, we can use it to generate estimates for CPU, memory, and disk storage, and to verify that we can build the cubes within the batch window. As an example, Figure 102 on page 202 shows the characteristics of a sample cube to estimate system requirements.

Cube 1

Number of dimensions – 7

- Member counts
- Dimension 1 – Dense – 34 members
- Dimension 2 – Dense – 42 members
- Dimension 3 – Dense – 3 members
- Dimension 4 – Sparse – 12 members
- Dimension 5 – Sparse – 134 members
- Dimension 6 – Sparse – 167 members
- Dimension 7 – Sparse – 1277 members

Percentage of maximum blocks existing 0.2%

Compression Ratio 0.6

Figure 102. Example cube criteria for hardware estimate

First, we calculate the number of blocks the cube will contain by multiplying together the member counts for the sparse dimensions, and then multiplying by the percent of maximum blocks existing:

(12 X 134 X 167 X 1277) X 0.2% = 685,841 -- (1)

Next, we calculate stored block size by multiplying together the member counts for the dense dimensions, multiplying by 8 bytes per value, and then applying the compression ratio:

((34 X 42 X 3) X 8) X 0.6 = 20.6 KB

Multiplying these values, we get an estimate of disk storage requirements:

685,841 X 20.6 = 14.1 GB disk storage

We get maximum parallelism when each cube is allocated its own storage devices. We must take this into account when totaling the disk requirements. If we have four cubes, each estimated at 3 GB, we should plan for 4 X 4 GB devices, rather than fewer larger devices.

Today's DB2 OLAP Server engine architecture is multi-threaded to support many concurrent users, but cube calculation is largely single-threaded. This means that the calculation process for one cube will drive one CPU. *Adding*

additional CPUs will not reduce the calculation time. To leverage SMP machines, we build a number of cubes in parallel, allocating one CPU to each cube.

Since we have a single cube, we require:

1 cube = 1 G6 processor

If you were able to conduct a Proof of Concept, then you will know how much data the machine was able to process. The key ratio we need is the number of gigabytes of data that the calculation generated per hour. Running on today's processors, with high-speed storage devices, such as IBM's Shark, we expect DB2 OLAP Server calculations to generate between 1 GB/hour and 3 GB/hour of data. Assuming we are running a G6 and dedicated Shark, we can calculate batch time by dividing estimated cube size by the data generation rate

14.1 / 3 = 4.7 hours batch time

We calculate cache memory for the index at (1) x 80 bytes per entry:

685,841 X 80 = 55 MB

We calculate cache memory for data blocks as three times the index cache, so total memory, including 2 MB overhead, is estimated for this cube as:

(55 X 4) + 2 = 222 MB memory

Based on these estimates, and allowing for error accordingly, our machine specification for this cube will be:

- 1 - G6 processor
- 250MB of memory
- 16 GB of disk storage
- With a batch window of 5 hours.

8.6 Summary

In this chapter we have discussed the key metrics we need to gather in order to make a good OLAP estimate. These metrics are extremely data dependent. Conducting a Proof of Concept using real data and building slices through the cube to gather the right information is the best way to get good estimates.

The information we need to gather is:

- Number of cubes
- Batch window
- Dimensions and member counts
- Sparse and dense dimension configuration
- Percentage of maximum blocks existing
- Dense block size
- Compression ratio
- Data generation rate during calculation

With this information, we can estimate CPU, memory and DASD. If the information is gathered as part of a Proof of Concept, then we can estimate to within 20 percent. Other estimating techniques are much less reliable.

Chapter 9. Security considerations

IBM DB2 OLAP Server is based on a client/server architecture that allows users to access enterprise data stored into OLAP “cubes”.

Therefore, it is very important to control the access to the OLAP data. DB2 OLAP Server for OS/390 benefits from the proven capabilities of OS/390 in the security area.

This chapter gives you an overview of the security architecture implemented on OS/390 and used by DB2 OLAP Server for OS/390 and its key components. It should help you to understand and to set up the right tasks for DB2 OLAP Server for OS/390 installation.

Figure 103 shows the components involved in DB2 OLAP Server security.

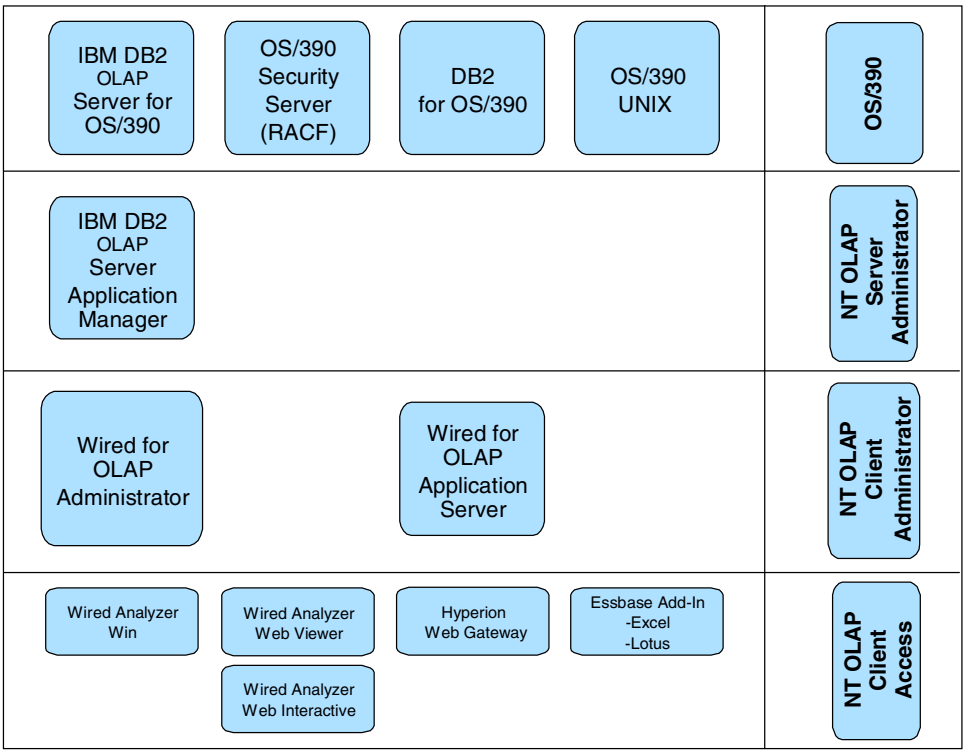


Figure 103. Components involved in DB2 OLAP Server security

To complete the installation, you need to set up the security requirements on the OS/390 layer (including OS/390 UNIX, DB2 for OS/390, RACF, and DB2 OLAP Server for OS/390), DB2 OLAP Server Administrator layer (including the Application Manager and Wired for OLAP Application Server - if installed in your environment), and DB2 OLAP clients.

In the OS/390 layer, you have to consider security options in several areas:

- Resource Access Control Facility (RACF)
- DB2 for OS/390
- OS/390 UNIX
- DB2 OLAP Server

There are many interactions between DB2 OLAP Server, DB2, and RACF.

Figure 104 on page 207 shows the various interactions between the components and should help you understand how DB2 OLAP Server works:

1. All the DB2 OLAP users must be defined to RACF. This includes the DB2 OLAP supervisor ID and the OLAP applications end-users.
2. You have to grant privileges to DB2 to the DB2 OLAP supervisor ID previously defined.
3. Define the OLAP applications users and their privileges to the DB2 OLAP Server. This can be done using the DB2 OLAP Server supervisor ID. You must specify to OLAP if you are defining a normal user or a new supervisor user.
4. During the previous definition, the DB2 OLAP Server does not involve RACF and does not perform any control. But if the user wants to connect to DB2 OLAP Server applications, that user ID must be defined to RACF.
5. Every end-user that connects to the DB2 OLAP Server must be defined to RACF and must provide a valid password. DB2 OLAP Server asks RACF to control the user and password validity.
6. When an end user accesses an OLAP application:
   a. The user sends the application request access to the DB2 OLAP engine.
   b. If the relational storage manager (RSM) is active, DB2 OLAP Server asks the DB2 subsystem to retrieve the data generating the appropriate SQL query to access the application star schema using the DB2 OLAP Server supervisor ID.

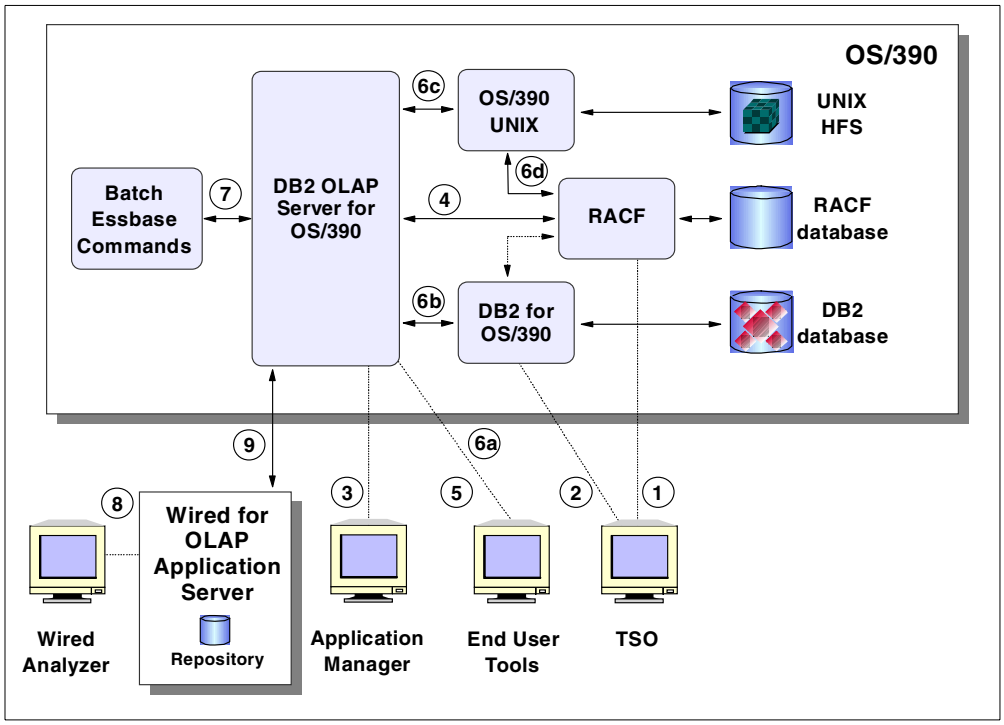


Figure 104. Security interactions between all OS/390 components

c. If the multidimensional storage manager (MDSM) is active, the DB2 OLAP Server engine accesses the HFS file system that handles the application data. The right UNIX permission bits should be set up to allow the DB2 OLAP Server to access the */app* application directory.

d. The access control to the */app* directory is done by OS/390 UNIX and RACF.

7. DB2 OLAP Server provides the capability to run batch commands using ESSCMD commands. These commands can be started from MVS batch JCL invoking the BPXBATCH program. The first command must be a login command to the DB2 OLAP engine using a valid end-user ID. RACF is involved in every login command.

8. End-users can access the DB2 OLAP Server using the Wired analyzer provided by the Wired for OLAP product. Each Wired for OLAP user must be defined as a DB2 OLAP Server user.

9. The Wired for OLAP Application Server provides a user mapping facility. You can define users to the Wired for OLAP repository in groups of users.

In this case, only the group user name has to be defined to the DB2 OLAP Server. All the access from the different users belonging to a group will be done under the user group name to the DB2 OLAP Server.

10. When the user of Wired Analyzer makes accesses to the OLAP database, the user ID and password should be provided to the DB2 OLAP Server.

9.1 Administering RACF security

To install IBM DB2 OLAP Server, you need to define two user IDs to RACF:

- The user ID used to install the DB2 OLAP Server code.
- The DB2 OLAP Server supervisor ID used to start the DB2 OLAP Server engine. This user ID will be used for the connection to the DB2 OLAP Server from the Application Manager. It enables the OLAP administrator to create other supervisor IDs needed to administer the DB2 OLAP Server.

You have to provide the DB2 OLAP Server supervisor ID and its password when you start the OLAP server.

For detailed information about how to define a user to RACF—interactive or batch—see 4.2.2, “Create a supervisor ID for DB2 OLAP Server” on page 57.

This is a summary of RACF definitions:

- Define the user ID and password of the DB2 OLAP Server installation user to RACF and DB2.
- Define the user ID and password of the supervisor to RACF, DB2, and then to DB2 OLAP Server.
- Define the user ID and password of normal users to RACF, then to DB2 OLAP Server.
- Groups defined in RACF and groups defined by DB2 OLAP Server Application Manager have nothing in common.

Verifying RACF password authorization

We strongly advise you to verify whether the user ID you use to login to the DB2 OLAP Server is defined to RACF by using the *PWDTEST* program. Refer to 4.2.7, “Verify RACF password authorization for DB2 OLAP Server” on page 64.

Important

If a user ID is not defined in RACF, the user cannot connect to the DB2 OLAP Server.

9.2 Administering DB2 security

If you decide to use the DB2 OLAP Server relational storage manager (RSM) DB2 security has to be set up for the DB2 OLAP Server supervisor ID to enable database and stogroup accesses.

By default, no user has the authority to access data in DB2, even if the user is defined to RACF. *However, all users do not have to be authorized by DB2 in order to run OLAP applications. Only the DB2 OLAP Server supervisor ID needs the appropriate DB2 privileges.*

DB2 must authorize the DB2 OLAP Server supervisor ID, and DB2 OLAP Server authorizes the user's or user group's privileges within OLAP Server integrated security. *So even though the user does not have DB2 authority, the user can read/write data, if DB2 OLAP Server security allows the user to do so.*

Give the DB2 authorities to:

- **The installation user ID:** needs the DB2 authority to create database, tablespaces, tables and the indexes for OLAP system administration and sample applications. Choose one:
  - SYSADM
  - CREATDBA
- **The DB2 OLAP Server supervisor ID**: needs the following DB2 authority to create tables and the indexes for OLAP applications and database administration:
  - DBADM authority for OLAP database or Create/Drop/Alter tables, views and indexes
  - Use of tablespace privilege on the OLAP tablespace
  - Use of stogroup privilege for index creation

9.3 Administering OS/390 UNIX security

For information about OS/390 UNIX security, refer to 3.5, “Prepare for OS/390 UNIX security” on page 47.

9.4 Administering DB2 OLAP Server security

Before starting to administer DB2 OLAP Server security, you should consider the following items:

- How many OLAP users are going to use the system?
- What applications are on the server?
- Which users will use which application?
- Which user should be allowed to access which data?

The objective is to structure users, applications, and data.

This section summarizes the security features provided by DB2 OLAP Server, and explains how to use them in the OS/390 environment. This includes:

- What structure exists to manage security in the DB2 OLAP Server.
- How to create, modify, and delete OLAP users or groups of OLAP users.
- How to use the security features of Essbase Application Manager.

For detailed information about DB2 OLAP Server security, refer to *Arbor Essbase Database Administrator’s Guide Volume 1,* SC26-9238.

Since the DB2 OLAP Server is a multiuser client/server system, security plays an important role. The DB2 OLAP Server provides different levels of security. It offers you the capability to implement the best security plan according to your environment. DB2 OLAP Server provides three different security layers:

- **Global access layer:** This layer lets you define privileges and access levels at the server, application, or database level. By default, all users have access (or no access) according to these global settings, unless they are granted higher privileges at the user and group layer.
- **User and Group layer:** This layers lets you define privileges and access levels for individual users or groups of users. These settings take precedence over global access settings. In this layer, you can define a user as a normal user or a supervisor.

- **Database filter layer**: This lets you define specific database access levels that individual users can have for particular database members, down to the individual data value (cell).

Various types of management privileges can be assigned to the global or user levels.

- Normal users have no management privileges.
- At the global layer, you assign management privileges on an application or database basis.
- At the user level, you assign management privileges to a specific user or a group, and its privileges for all applications and databases.

How to administer security

When you begin to administer security, you can use ESSCMD or Essbase Application Manager. The following steps are needed to administer security from the Application Manager:

- Click **Start--> Programs--> IBM DB2 OLAP Server 1.1--> Essbase Application Manager**
- Click on **Server--> Connect** of the main menu.
- In the Essbase System Login window enter:
  - the OLAP Server name or its TCP/IP address
  - User name and password
  - Click **OK** to connect to the OLAP Server

Note

In order to administer security for the first time, you must use the user ID and password of the DB2 OLAP Server supervisor that is provided at startup.

- After login, click on **Security** to administer security in the Application Manager window.
- Choose each subtask of the Security bar to specify security requirements as shown Figure 105 on page 212. For detailed steps about how to create a user, how to authorize an application to the user, and how to authorize access of a database to the user, refer to *Hyperion Essbase Database Administrator's Guide Volume 1,* SC26-9238.

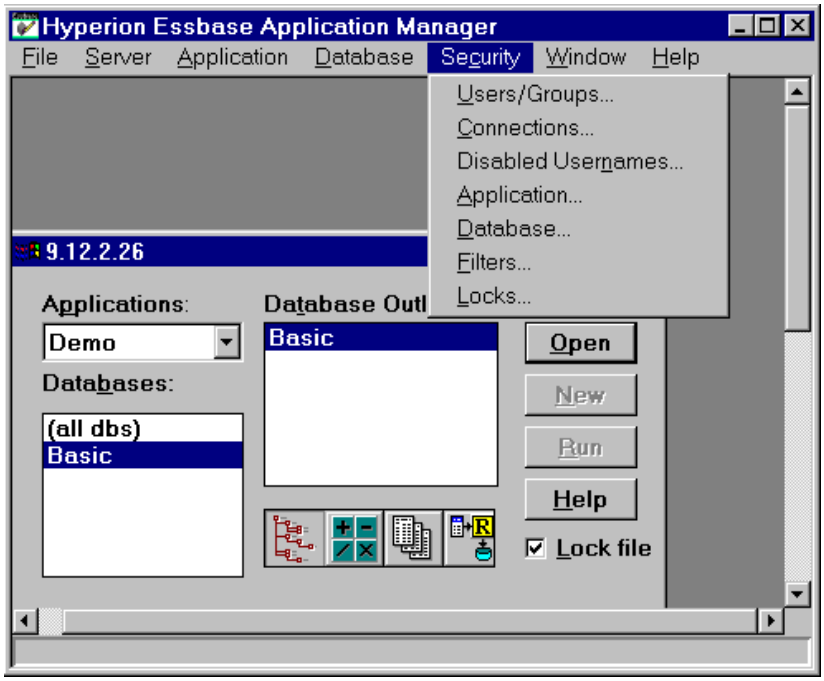


Figure 105. Security options on Application Manager

9.4.1 Administering global level privileges

The global access layer pertains directly to the security-access settings for applications and databases and their related files. Application and database security settings are based on the *minimum database access* privilege granted to all users.

Users with *supervisor* privilege, *application designer* privilege for the application, or *database designer* privilege for the database are not affected by these settings. Supervisors automatically have full access, and application designers and database designers have full access only for their applications or databases.

By default, users and groups inherit the global access settings, which become their security privileges. A user can, however, have application and database privileges that go beyond the global defaults.

The access privileges in Table 12 are available in the global access layer. These privileges apply to applications and databases.

Table 12. Access level options for application and databases

Access level	Privilege description
None	Specifies *no access* to any file or data value in the application or database. This is the default access privilege assigned when an application or database is created.
Read	Specifies *read-only* access to any object or data value in the application or database. Users can view files, retrieve data values, and run report scripts.
Write	Specifies *update* access to any data value in the application or database. Users can view Essbase files, retrieve and update data values, and run report scripts. Write access does not permit calculations or outline modifications.
Calculate	Specifies *calculate and update* access to any data value in the application or database. Users can view files; retrieve, update, and perform calculations based on data values; and run report and calc scripts. Calculate access does not permit outline modifications.
Database designer	In addition to *calculate and update* access, DB designer access lets users view and modify the outline.

9.4.1.1 Defining global application access

You can define access settings and other settings that apply to applications on a global level. The settings you define for the application affect all users, unless they have higher privileges granted to them at the user level. The following application settings are available:

- A brief description of the application.
- A time-out setting for locks.
- Options that control application loading, command processing, connections, updates, and security.
- Settings that define the minimum access level to databases within the application.

Only users with supervisor privilege (or application designer privilege for the application) can change global access settings for applications.

To define settings for an application, choose **Application--> Settings** in Application Manager.

Minimum database access
This means that all databases within the application inherit the settings specified in the application settings unless they are changed at the database level. There are five access options, described in Table 12 on page 213.

Allow settings
These settings override the security privileges of other supervisors. When a supervisor clears any of the check boxes, all other supervisors are affected by the change. If you are new to DB2 OLAP Server or Hyperion Essbase, we recommend you not change these settings.

9.4.1.2 Defining global database access
When you create a database, it inherits the global access settings for the application. However, any database within an application can be defined with its own global access setting, which will override the application global settings.

To define settings for a database, choose **Database--> Settings** in Application Manager.

9.4.1.3 Assigning global access settings per user
Users and groups can be assigned *application designer* or *database designer* privilege on an application or database basis. These settings are useful for assigning supervisory privileges to users who need to be in charge of particular applications or databases, but who only need normal user privileges for other applications.

Application designer privilege
If you have application designer privilege for an application, you have complete access to all objects in that application. You, however, cannot create or delete an application unless you also have been granted that privilege on the user level. With application designer privilege, you can:

- Modify any object within the application
- Create, modify, and delete databases within the application
- Assign user access privileges at the application or database level for the application
- Define and assign filter objects anywhere in the application
- Remove data locks within the application
- Disconnect users who are connected to the application or any application for which you have application designer privilege

Application designer privilege applies only to the assigned application. Outside of the application, you revert to the privileges of an ordinary user.

With *Create/Delete Apps* privilege, you are automatically given application designer privilege for any application you create.

Database designer privilege
If you have database designer privilege, you have complete access to all objects in the database. You cannot create or delete a database, but you can:

- Modify any object within the database
- Restrict data access privileges at the database level for the database
- Define and assign filter objects anywhere in the database
- Remove data locks within the database

Database designer privilege applies only to data access for the assigned database. Outside of the database, you revert to the privileges of an ordinary user.

9.4.2 Administering users and groups

In addition to global access settings, the user or group layer lets you define security settings for individual users and user groups. Groups are collections of users that share the same minimum privileges. Users inherit all privileges of the group and can have additional access to privileges that exceed those of other group members. When you create a new user, you can assign the user to a group. A user can be member of one group or more than one group. But every user does not have to be a member of a group.

The process for creating, editing, or copying groups is the same as that for users, except that there is no group password.

9.4.2.1 Type of users or groups

When you create or edit the users and groups, you can assign privileges by defining *type* of users and groups, which includes *supervisor* or *user*.

1. Supervisor privilege has:
   - Full access to all applications, databases, related files, and security mechanisms for a server.
   - Privileges to create, delete, edit, and rename all users, including other supervisors.
   - Privileges to create and delete applications.

   The user is not affected by any global access settings.

2. A user with user privilege is an ordinary user, who can:
   - Access only those objects to which the user ID has been granted privileges. This user needs more privileges if wider access is needed.
3. A user with Create/Delete Users/Groups privilege can:
   - Create and edit users with equal or lower privileges only.
   - Delete or rename users regardless of their privileges.
   - Rename users regardless of their privileges.
   - Create, edit, delete, or rename groups with equal or lower privileges only.
4. A user with Create/Delete Applications privilege can:
   - Create, modify, and delete applications and databases.
   - Assign user access privileges to applications and databases at the Global Access layer.
   - Define and assign filter objects.
   - Remove data locks.

   Users with Create/Delete Applications privilege cannot create or delete users, but they can manage global access to those applications which they have created.

9.4.2.2 Managing users and groups

There are five user-management tasks available at varying degrees to differently privileged users. To begin managing users and groups, you can use ESSCMD or Application Manager; see “How to administer security” on page 211.

1. Create new users and groups:
   - When you create a new user, use a valid RACF user ID with password. At definition time, user ID and password are not validated by DB2 OLAP Server and RACF. But when you connect to DB2 OLAP Server to access data, the user ID and the password will be validated by RACF.
   - The password is ignored at the DB2 OLAP Server side. Ignore *the User must change the password at the next login* button in Application Manager.
   - Assign the user type as either Supervisor or User.

2. Edit (modify the privileges of) existing users and groups:

   Editing a user means to modify the security profile established when the user was created. Any privilege or limitation that is not assigned when creating a new user can be modified using the *Edit User* capability.

   - Changing the user password is not allowed. (This is possible only with RACF.)
   - Change the user's group membership, the user's application access, the user type.
   - Lock or unlock the user 's access to the server for any reason. Only a system administrator with supervisor privilege can change this option.

3. Copy the security profiles of existing users and groups.
4. Delete users and groups:
   - Even though you delete a user from DB2 OLAP Server, the user still exists in the RACF database. You can define this user again without redefining the user to RACF.
   - After deleting the user, the user is automatically deleted from all groups to which it belonged.
5. Rename users and groups:
   - You can rename any user ID. At definition time, DB2 OLAP Server does not validate the ID. It will be validated at database connection time.

9.4.2.3 Assigning application and database access to a user

By default, users and groups inherit the global application and database settings, which become their security privileges. A user can, however, have application and database privileges that go beyond the global defaults. These settings can be defined by a system administrator when creating a new user or editing an existing user.

This setting is not needed for the supervisors because they automatically have full privileges to every application on the server.

To modify application or database access settings for a user:

1. Go to **Security--> Users/Group** from Application Manager.
2. Select a user name, then click **Edit User**.
3. The User/Group application access have three options:
   - None: The user has no access to the selected application or any of its databases.

- Access DBs: You need to define the specific access privileges for each database within the application. Refer to 9.4.2.4, “Assigning database access to a user” on page 218.
- Application designer: This option gives the user full access to the application. See, “Application designer privilege” on page 214.

9.4.2.4 Assigning database access to a user

There is no need to assign database access for supervisors, or for those with *application designer* privilege for the application or *database designer* privilege for the database. These users already have full database access. You need to assign database access to other users if:

- The users have not been granted sufficient user privileges to access databases.
- The database in question does not allow users sufficient access through its global settings.
- The users have no access granted to them through filters.

When you assign access to databases within the selected application, proceed as follows:

1. Go to **Security--> User/Groups** from Application Manager.
2. Select a user, and then click **Edit user--> App Access**.
3. Check **Access DBs** and click **DB Access**. The DB Access button then becomes available.
4. You can assign one access option based on the user’s requirements and the database characteristics:

Table 13. Description of database access level

User Access level	Privilege description
None	Indicate no access to any object or data value in a database.
Filter access	Indicate that data access is restricted to those filters assigned to the user.
Read only	Indicate read access to retrieve all data values. Report scripts can also be run.
Read/write	Indicate that all data values can be retrieved and updated (not calculated). The user can run, but can not modify Essbase objects.
Calculate	Indicate that all data values can be retrieved and updated and calculated with the default outline or any calc scripts to which user has access. See note 1.
Database designer	Indicate that all data values can be retrieved and updated and calculated. In addition, all database-related files can be modified.
Filter	Associates a filter object with a user name. A user can have one filter per database.

Note 1) Any calculation scripts defined afterward are automatically added to the user's calculate privileges. Individual calculation script privileges can be added or removed.

By default, a supervisor, application designer, or database designer can run all calc scripts.

9.5 Administering DB2 OLAP client security

Any client that would access data in the DB2 OLAP Server needs to be authorized by:

- RACF for the user ID and the password
- DB2 OLAP Server for a user ID (same user ID as RACF)
- DB2 OLAP Server for accessing OLAP applications and the databases

9.5.1 Add-in features

Any client user who is using Lotus 1-2-3 or Microsoft Excel can access an OLAP database by using the Add-in feature. You need:

- A valid RACF user ID and password

- A valid user ID of DB2 OLAP Server, which is same as the RACF user ID
- Authorization to access the OLAP applications and the databases

9.5.2 Hyperion Wired for OLAP

Without a valid user ID, no one is able to use DB2 OLAP Server applications, and this is also true for Wired for OLAP applications. Prior to administering security with Wired for OLAP applications, you have to understand the structure of the Hyperion Wired for OLAP, and the relationship between Wired for OLAP and DB2 OLAP Server. Wired Administrator is the tool that manages who is allowed to look at what kind of data within the Wired for OLAP environment.

User management
There are two types of users in Wired for OLAP:

- User: This is the default setting
- System manager: This user is assigned additional privileges and responsibilities with regard to system administration, such as:
  - Utilize Wired Administrator tools
  - Automatically distribute lists
  - Relocate Wired repositories
  - Define virtual cubes in relational databases
  - Create custom graphic displays using Wired Designer (pins, pinboards, and forms)

User group management
User groups allow users with similar needs to be grouped. Their user profiles can be addressed and edited cooperatively, that makes the administration of users easy and quick. You define the user group name only.

Database connection management
A user or user group needs to be connected to each OLAP database to build the reports or charts using Wired for OLAP applications. The database connections and database properties are maintained in the Wired repository.

When you access a DB2 OLAP database using a Wired for OLAP application, you need two user IDs to:

- Login to Wired for OLAP: This ID is stored in Wired repository and is valid within Wired for OLAP only. The Wired administrator creates a user ID and password.

- Access to databases in the DB2 OLAP Server: After choosing the application and its database from the Wired for OLAP application, you should provide a user ID and password to access the application and its database. This user ID and password must be valid in RACF.

  There are three ways to provide a user ID and password from Wired users when the database connection is required.

  - Use Wired user ID and password: This method works only when the Wired user ID and password and the DB2 OLAP Server user ID and password are identical.
  - Prompt for user ID and password: This method makes Wired prompt the user when the database connection is established.
  - Assign authorized user ID and password: This method enables Wired to automatically reference this user information in future connections to the database. It should be an appropriate DB2 OLAP Server user ID.

Wired Administrator

You can use the Wired Administrator to administer security within Wired for OLAP. It is delivered as part of the Wired Analyzer professional edition and provides a graphical interface.

You can start the Wired Administrator application by:

1. Clicking **Programs--> Hyeprion Wired for OLAP--> Wired Administrator**, or Double-clicking the Wired Administrator file name (Admin) in your *Wireddirectory.*
2. Login with a system-provided administrator user ID and password and then the Wired administrator window appears as shown in Figure 106.

Note

After installation of Hyperion Wired for OLAP, the user ID=*system* password=*12345* with supervisor privileges is created. In order to prevent uncontrolled administrator access, you should change the password as soon as possible.

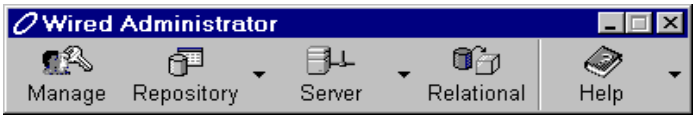


Figure 106. Wired Administrator window

Among the Wired Administrator icons, the *Manage* icon opens the management window, enabling:

- Wired user profile management
- User group management
- Database connection management

System setting

The Wired Administrator may choose either NT Domain authentication or Hyperion Essbase Security to authenticate logins.

- NT domain authentication

  If this option is selected, Wired for OLAP checks to see if a Windows NT user is logged into a domain. If so, Wired tries to log the user in using the NT user name.

- Hyperion Essbase security

  When *use of Hyperion Essbase Security* is enabled, Wired for OLAP uses the specified Essbase server to validate user ID, password, and user type for login; see Table 14.

Table 14. Hyperion Essbase security algorithm

If Hyperion Essbase ...	**Then Wired for OLAP ...**
validates the user and a Wired user account exists	logs in the user.
validates the user but can not validate the Wired user profile	creates a new user based on the user settings in Wired Administrator default user ID. Assigns the Hyperion Essbase user ID, password, and access privileges to the new user account. Hyperion Essbase users with supervisor privileges are assigned system manager privileges in Wired.
cannot validate the user	Attempts to log the user in using the Wired for OLAP security in the SYSTEM Manager-created user profile.

Note

To avoid confusion when maintaining user accounts, it is recommended that you do not mix user accounts created in Wired Administrator and user accounts that were created using Hyperion Essbase Security.

If you already have user accounts created in Wired Administrator but want to enable the Hyperion Essbase feature, it is recommended that you delete all user accounts (except user ID *system*; password *12345*) that were created in Wired Administrator. Enable Hyperion Essbase Security, and let Essbase create new users as they log into Wired for OLAP.

9.5.3 Controlling user access on the Web server

To ensure the best possible security for your data, choose a server with a secure protocol, such as the Netscape Commerce Server or the Microsoft Merchant Server. Secure protocols encrypt sensitive data, such as passwords.

You can also limit access to the esscgi.exe program by keeping it in just one directory with your other executable files, instead of copying esscgi.exe to each subdirectory.

9.5.4 Controlling user access in the Web Gateway application

The Essbase Web Gateway (EWG) Application Designer has three options for controlling user access within the application:

- Design the Login template to restrict access
- Provide read-only or update access
- Use other ways to control access

Design the login template to restrict access

The Login template gives you many combinations of options to control user access. You can give users the ability to enter the user name, server, and password; you can provide defaults for that information, which users can overwrite; or you can hard-code that information. You can also define the application and server that a certain user name connects to. You can design many Login templates for different access methods.

Provide read-only or update access

The Retrieve template can include an Update Mode button at the Application Designer's discretion. The EssControls tag provides a set of default options. If

you include the EssControls tag in the Retrieve template, an Update Mode button appears automatically. When users click the Update Mode button, EWG displays the Update template, which gives users read/write access to data.

If you want users to have read-only access to data in the Retrieve template, do not use the EssControls tag; specify the control tags you want to use.

Other ways to control access

In the Retrieve and Update templates, you can choose what actions users can perform (for example, zoom in or pivot). The EssControls tag provides a set of default options. To restrict users to a certain set of actions, do not use the EssControls tag in the Retrieve or Update templates. Instead, use the EssDoAction tags, where Action is a type of viewing option for data, such as Zoom In.

Appendix A. National language environment variables

Table 15 contains a list of supported client code set and server locale combinations for each national language.

Table 15. Environment variables for national languages

Language	Environment variable settings
United States English	DB2OLAP_CLIENT_CODESET=”ISO8859-1” GC_LANG=”English_UnitedStates.IBM1047” LC_ALL=En_US.IBM-1047
Brazilian Portuguese	DB2OLAP_CLIENT_CODESET=”IBM-850” GC_LANG=”Portuguese_Brazil.IBM1047” LC_ALL=Pt_BR.IBM-1047
Czech	DB2OLAP_CLIENT_CODESET=”IBM-852” GC_LANG=”Czech_CzechRepublic.IBM870” LC_ALL=Cs_CZ.IBM-870
United Kingdom English	DB2OLAP_CLIENT_CODESET=”ISO8859-1” GC_LANG=”English_UnitedKingdom.IBM1047” LC_ALL=En_GB.IBM-1047
French	DB2OLAP_CLIENT_CODESET=”IBM-850” GC_LANG=”French_France.IBM1047” LC_ALL=Fr_FR.IBM-1047
German	DB2OLAP_CLIENT_CODESET=”IBM-850” GC_LANG=”German_Germany.IBM1047” LC_ALL=De_DE.IBM-1047
Hungarian	DB2OLAP_CLIENT_CODESET=”IBM-852” GC_LANG=”Hungarian_Hungary.IBM870” LC_ALL=Hu_HU.IBM-870
Italian	DB2OLAP_CLIENT_CODESET=”IBM-850” GC_LANG=”Italian_Italy.IBM1047” LC_ALL=It_IT.IBM-1047
Japanese	DB2OLAP_CLIENT_CODESET=”IBM-932” GC_LANG=” Japanese_Japan.IBM939” LC_ALL=Ja_JP.IBM-939
Korean	DB2OLAP_CLIENT_CODESET=”IBM-949” GC_LANG=” Korean_Korea.IBM933” LC_ALL=Ko_KR.IBM-933

Language	Environment variable settings
Russian	DB2OLAP_CLIENT_CODESET=”IBM-866” GC_LANG=”Russian_Russia.IBM1025” LC_ALL=Ru_RU.IBM-1025
Simplified Chinese	DB2OLAP_CLIENT_CODESET=”IBM-936” GC_LANG=”Simplified Chinese_China.IBM935” LC_ALL=Zh_CN.IBM-935
Spanish	DB2OLAP_CLIENT_CODESET=”IBM-850” GC_LANG=”Spanish_Spain.IBM1047” LC_ALL=Es_ES.IBM-1047
Traditional Chinese	DB2OLAP_CLIENT_CODESET=”IBM-950” GC_LANG=”TraditionalChinese_Taiwan.IBM937” LC_ALL=Zh_TW.IBM-937

DB2 OLAP Server for OS/390 product users that use the Korean language will not be able to refer to user-defined characters in the range 0xDA41—0xDDFD.

Appendix B. REXX procedure to create TCP/IP affinity

This REXX procedure creates TCP/IP affinity between a TSO user and a TCP/IP stack. As shown in Figure 107 and Figure 108 on page 228, in this example, we have three types of stacks:

```
/* REXX */
parse upper arg stack .
sysname = mvsvar('sysname')
op_sys = mvsvar(sysopsys)
op_sys_words = words(op_sys)
fmid = word(op_sys,op_sys_words)
if stack = 'OE' | stack = 'OMVS' | stack = 'USS' then
  do
    if fmid = 'JBB6607' | fmid = 'HBB6608' then
      do
        tcpdata = "'tcpipoe."||sysname||".tcpparms(tcpdata)'"
        ftpdata = "'tcpipoe."||sysname||".tcpparms(ftpdata)'"
      end
    else
      do
        tcpdata = "'tcpipoe."||sysname||".tcpip.data'"
        ftpdata = "'tcpipoe."||sysname||".ftp.data'"
      end
    "alloc f(systcpd) da("tcpdata") shr reuse"
    "alloc f(sysftpd) da("ftpdata") shr reuse"
  end
```

Figure 107. Sample REXX procedure (1/2)

- OE: This is a TCP/IP stack that can be used from a telnet session.
- OMVS: This is a TCP/IP stack that can be used from a TSO OMVS session.
- USS: This is a TCP/IP stack that can be used from OS/390 UNIX. Basically this is same as an OE stack.

```
else
  if stack = 'MVS' | stack = 'TSO' then
    do
      if fmid = 'JBB6607' | fmid = 'HBB6608' then
        do
          tcpdata = "'tcpipmvs."||sysname||".tcpparms(tcpdata)'"
          ftpdata = "'tcpipmvs."||sysname||".tcpparms(ftpdata)'"
        end
      else
        do
          tcpdata = "'tcpipmvs."||sysname||".tcpip.data'"
          ftpdata = "'tcpipmvs."||sysname||".ftp.data'"
        end
      "alloc f(systcpd) da("tcpdata") shr reuse"
      "alloc f(sysftpd) da("ftpdata") shr reuse"
    end
  else
    if stack = 'NONE' then
      do
        "free f(systcpd,sysftpd)"
      end
    else
      say 'Operand must be OE, OMVS, USS, MVS or TSO, or NONE'
exit 0
```

Figure 108. Sample REXX procedure (2/2)

Appendix C. Summary of DB2 OLAP Server files

Table 16 lists the DB2 OLAP Server product main installation files and directories.

Table 16. DB2 OLAP Server main installation files and directories

DB2 OLAP Server files or directories	Description
/usr/lpp/essbase/tar	DB2 OLAP Server tar directory
/usr/lpp/essbase/tar/DB2OLAP.Z	Product tar file
/usr/lpp/essbase/tar/setup.sh	Installation script
/usr/lpp/essbase/tar/390/server/essinst	Server installation program
/usr/lpp/essbase/tar/390/server/server.Z	Server installation code
/usr/lpp/essbase/tar/390/server/samples1.Z	Sample applications code
/usr/lpp/essbase/tar/390/api/essinst	API installation program
/usr/lpp/essbase/tar/390/api/api.Z	API installation code
/usr/lpp/essbase/tar/docs	Empty docs directory on CD
/etc/essbase	DB2 OLAP Server configuration directory
/etc/essbase/dsnaoini	DB2 CLI configuration file (text file)
/etc/essbase/rsm.cfg	RSM configuration file (text file)
/u/essbase	DB2 OLAP Server home directory
/u/essbase/essbaseenv.doc	Environment variables
/u/essbase/bin	DB2 OLAP Server executables directory
/u/essbase/bin/essbase.cfg	DB2 OLAP Server Configuration file (text file)
/u/essbase/app	Application data and samples directory
/u/essbase/app/_tmpinst	Sample source files directory
/u/essbase/locale	NLS translate tables directory
/u/essbase/api	API files directory

Table 17 lists directories and files created by the DB2 OLAP Server to store application information and data. If using RSM, application data files and index files are stored in a DB2 relational database.

We assume in this list that the DB2 OLAP Server home directory is /u/essbase. Myapp is defined as an OLAP application name and Mydb as an OLAP application database.

Table 17. DB2 OLAP Server main application files and directories

Application files or directories	Description
/u/essbase/app/Myapp	Application directory
/u/essbase/app/Myapp/Myapp.app	Binary description of application
/u/essbase/app/Myapp/Myapp.log	Application log file (text file)
/u/essbase/app/Myapp/Mydb	Database directory
/u/essbase/app/Myapp/Mydb/Mydb.otl	Database outline (binary file)
/u/essbase/app/Myapp/Mydb/Mydb.log	Database change log
/u/essbase/app/Myapp/Mydb/Mydb.db	Binary description of database
/u/essbase/app/Myapp/Mydb/Mydb.dbb	Backup of db binary description
/u/essbase/app/Myapp/Mydb/Mydb.ddb	Partitioning definitions (binary file)
/u/essbase/app/Myapp/Mydb/Mydb.ddm	Temporary partitioning file
/u/essbase/app/Myapp/Mydb/Mydb.ddn	Temporary partitioning file
/u/essbase/app/Myapp/Mydb/Mydb.esm	Storage manager file
/u/essbase/app/Myapp/Mydb/Mydb.esn	Temporary storage manager file
/u/essbase/app/Myapp/Mydb/Mydbl.rul	Load rule stored at server (binary file)
/u/essbase/app/Myapp/Mydb/Mydb.rep	Report script (text file)
/u/essbase/app/Myapp/Mydb/Mydb.tct	Database transaction control file
/u/essbase/app/Myapp/Mydb/Mydb.txt	Text file to load
/u/essbase/app/Myapp/Mydb/Mydbc.csc	Calc script stored at server (text file)
/u/essbase/app/Myapp/Mydb/ess00001.ind	Application block index file for MDSM
/u/essbase/app/Myapp/Mydb/ess00001.pag	Application data file for MDSM

Appendix D. Special notices

This publication is intended to help OS/390 technical specialists and DB2 OLAP Server basis consultants to implement the DB2 OLAP Server for OS/390 Version 1.1. The information in this publication is not intended as the specification of any programming interfaces that are provided by DB2 OLAP Server for OS/390. See the PUBLICATIONS section of the IBM Programming Announcement for DB2 OLAP Server for OS/390 Version 1.1 for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

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, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate

them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

CBPDO	DB2 Connect
DB2	DB2 OLAP Server
DFSMS/MVS	DFSMSdss
IBM	OpenEdition
OS/390	QMF
RACF	SMP/E
S/390	System/390

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere.,The Power To Manage., Anything. Anywhere.,TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and/or other countries licensed exclusively through The Open Group.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Appendix E. Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

E.1 International Technical Support Organization publications

For information on ordering these ITSO publications see “How to get ITSO redbooks” on page 237.

- *OS/390 Version 2 Release 6 UNIX System Services Implementation and Customization,* SG24-5178
- *Getting Started with Data Warehouse and Business Intelligence,* SG24-5415
- *Managing Multidimensional Data Marts with Visual Warehouse and DB2 OLAP Server,* SG24-5270
- *DB2 UDB for OS/390 Version 6 Performance Topics,* SG24-5351
- *Getting Started with ADSM: A Practical Implementation Guide*, SG24-5416
- *TCP/IP Tutorial and Technical Overview*, SG24-5227

E.2 Redbooks on CD-ROMs

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at http://www.redbooks.ibm.com/ for information about all the CD-ROMs offered, updates and formats.

CD-ROM Title	Collection Kit Number
System/390 Redbooks Collection	SK2T-2177
Networking and Systems Management Redbooks Collection	SK2T-6022
Transaction Processing and Data Management Redbooks Collection	SK2T-8038
Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
AS/400 Redbooks Collection	SK2T-2849
Netfinity Hardware and Software Redbooks Collection	SK2T-8046
RS/6000 Redbooks Collection (BkMgr)	SK2T-8040
RS/6000 Redbooks Collection (PDF Format)	SK2T-8043
Application Development Redbooks Collection	SK2T-8037
IBM Enterprise Storage and Systems Management Solutions	SK3T-3694

E.3 Other publications

These publications are also relevant as further information sources:

- *DB2 OLAP Server Start Here,* GC26-9236
- *DB2 OLAP Server Installation Notes,* GC26-9237
- *Using DB2 OLAP Server,* SC26-9235
- *DB2 OLAP Server DB Administrator's Guide Volume 1,* SC26-9238
- *DB2 OLAP Server DB Administrator's Guide Volume 2,* SC26-9286
- *Spreadsheet Add-in User's Guide for Excel,* SC26-9240
- *Spreadsheet Add-in User's Guide for 1-2-3,* SC26-9242
- *SQL Drill-Through Guide,* SC26-9244
- *DB2 OLAP Server Quick Technical Reference,* SC26-9239
- *DB2 UDB ODBC Guide and Reference,* SC26-9005
- *DB2 UDB Utility Guide and Reference,* SC26-9015
- *DB2 Call Level Interface Guide and Reference,* SC26-8959
- *OS/390 UNIX System Services Planning,* SC28-1890
- *OS/390 eNetwork Communications Server: IP Configuration*, SC31-8513
- *MVS Setting Up a Sysplex*, GC28-1779

E.4 Referenced Web sites

These Web sites are also relevant as further information sources:

- `http://www.hyperion.com/` Hyperion home page
- `http://www.olapcouncil.org/` OLAP Council home page

How to get ITSO redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** http://www.redbooks.ibm.com/

  Search for, view, download, or order hardcopy/CD-ROM redbooks from the redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this redbooks site.

  Redpieces are redbooks in progress; not all redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

  Send orders by e-mail including information from the redbooks fax order form to:

	e-mail address
In United States	usib6fpl@ibmmail.com
Outside North America	Contact information is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the redbooks Web site.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and redbooks by accessing the IBM Intranet Web site at http://w3.itso.ibm.com/ and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access MyNews at http://w3.ibm.com/ for redbook, residency, and workshop announcements.

IBM Redbook fax order form

Please send me the following:

Title	Order Number	Quantity

First name	Last name	
Company		
Address		
City	Postal code	Country
Telephone number	Telefax number	VAT number

☐ Invoice to customer number _____

☐ Credit card number _____

Credit card expiration date	Card issued to	Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Glossary

Accounts dimension. A dimension type that makes accounting intelligence available. You can tag one dimension as an accounts dimension, but an accounts dimension is not required.

Administrator. A person who is responsible for installing and maintaining the DB2 OLAP Server and for setting up user accounts and security.

Agent. A process that starts and stops applications and databases, manages connections from users, and handles user-access security.

Alias ID table. A table created by DB2 OLAP Server in your relational database that contains a mapping of Essbase alias table names to ID numbers allocated by the DB2 OLAP Server.

Alias ID view. A view created by the DB2 OLAP Server in your relational database that contains one row for each Essbase alias table used with a relational cube. There is one alias ID view for each relational cube.

Alias name. An alternate name for a dimension or member.

Anchor dimension. A dense dimension that is specified as the dimension that the DB2 OLAP Server uses to help define the structure of the fact table it creates in the relational cube for an Essbase database.

API. Application programming interface. The Essbase API is a library of functions that you can use in a custom C or Visual Basic program to access the DB2 OLAP Server.

Application section. A section in the Relational Storage Manager configuration file that contains values for parameters that override the values for the same parameters in the RSM section of the configuration file.

Block. A string of data elements recorded or transmitted as a unit.

Cache. A component of memory. Each Essbase database contains a data cache and an index cache.

Calculation. An equation within a database outline, a calculation script, or a report script that calculates a value for a particular member or point in a report.

Calculation script. A text file that contains instructions to perform calculations within an Essbase database. Also called a calc script.

Commit Block parameter. A parameter on the Transaction page of the Database Settings notebook in the Essbase Application Manager that you use to set the number of blocks that can be changed before the DB2 OLAP Server commits the blocks.

Cube catalog table. A table that the DB2 OLAP Server creates in your relational database that contains a list of all the Essbase databases that are stored in your relational database. The cube catalog table also shows the application with which each cube is associated. Each time that you create an Essbase database, the DB2 OLAP Server creates a new row in this table.

Cube catalog view. A view that the DB2 OLAP Server creates in your relational database that allows an SQL user to access a list of Essbase applications and relational cubes.

Cube table. A table that the DB2 OLAP Server

creates in your relational database that contains a list of dimensions in a relational cube and information about each dimension.

Cube view. A view that the DB2 OLAP Server creates in your relational database that allows an SQL user to access the names of all dimensions in a relational cube and associated information for each dimension. There is one cube view for each relational cube in your relational database.

Data load. The process of populating an Essbase database with data. Loading data establishes actual values for the values of the cells defined in the database outline for the database.

Data load rules. A set of operations that the DB2 OLAP Server performs on data as it is loaded from an external source file.

Database administrator. A person responsible for administering a relational database.

Database log file. A set of primary and secondary log files consisting of log records that record all changes to a database. The database log file is used to roll back changes for units of work that are not committed and to recover a database to a consistent state.

Database managed space (DMS). Space in a table that is managed by the database.

Database name. The name of the relational database where you want the DB2 OLAP Server to store your multidimensional data.

Database outline. The structure that defines all elements of a database within Essbase. It contains definitions of dimensions and members, dense or sparse dimension tags and attributes, the anchor dimension attribute, calculations, shared members, and alternations to the basic roll-up structure of the database.

Database password. The password for the user ID that you want the DB2 OLAP Server to use to log on to your relational database.

Database section. A section in the Relational Storage Manager configuration file that contains values for parameters that override the values for the same parameters for the current application.

Database settings. Settings that you can modify to improve performance and space utilization for your relational database. You can change database settings by using utilities or commands supplied with your relational database management system.

Database user ID. The user ID that you want the DB2 OLAP Server to use to log on to your relational database. The default is the supervisor ID that you specify when you start the DB2 OLAP Server for the first time.

Dense dimension. A dimension with a high probability for occupying one or more data points in every combination of dimensions that occurs.

Dimension. A data category, such as time, accounts, products, or markets. In an Essbase database outline, the dimensions represent the highest consolidation level.

Dimension table. A table that the DB2 OLAP Server creates in your relational database that contains detailed information about the members in a dimension. There is one dimension table for each dimension in an outline.

Dimension view. A view that the DB2 OLAP Server creates in your relational database that allows an SQL user to access information about members contained in a dimension.

Essbase API. A library of functions that you can use in a custom C or Visual Basic program to access the DB2 OLAP Server.

Essbase application. An application that you create using the Essbase Application Manager or Essbase commands. An Essbase application can contain one or more Essbase databases and any associated calculation scripts, report scripts, and data load rules.

Essbase Application Manager. A tool that you can use to create and maintain Essbase applications.

Essbase database. A multidimensional database that you create using the Essbase Application Manager or Essbase commands. An Essbase database includes a database outline, data, associated optional calculation scripts, optional report scripts, and data load rules. The DB2 OLAP Server stores the actual data and a shadow of the database outline in tables in a relational database.

Essbase Spreadsheet Add-in. Software that merges seamlessly with Microsoft Excel and Lotus 1-2-3. The software library appears as a menu Add-In to the spreadsheet and provides such features as connect, zoom-in, and calculate.

ESSCMD. A command-line interface used to perform server operations interactively or through a batch file.

Fact table. A table, or in many cases a set of four tables, that the DB2 OLAP Server creates in your relational database that contains all data values for a relational cube.

Fact view. A view that the DB2 OLAP Server creates in your relational database that allows an SQL user to join fact data to dimensions to access the actual data values in a relational cube.

Generation name. A unique name that describes a generation in a database outline.

Generation table. A table that the DB2 OLAP Server creates in your relational database that contains generation numbers and names for each named generation specified when you created the outline. There is one generation table for each dimension in an outline.

Isolation level. A parameter that determines how data is locked or isolated from other transactions and processes while the data is being accessed. You can set the isolation level in the Relational Storage Manager configuration file (RSM.CFG).

Key table. A table that the DB2 OLAP Server creates in your relational database that is equivalent to the Essbase Index. The DB2 OLAP Server creates the key table after the first successful restructure.

Level name. A unique name that describes a level in a database outline.

Level table. A table that the DB2 OLAP Server creates in your relational database that contains level numbers and names for each named level specified when you created an outline. There is one level table for each dimension in an outline.

Member. A discrete component within a dimension. For example, January 1997 or 1Qtr97 are typical

members of a Time dimension.

Multidimensional data. The data in an Essbase database. Data can include basic data values (loaded from an external source) that represent combinations of the lowest level of members in the dimensions of the database; data values that are calculated from the base data values; and rolled up data values that are created by combining values for members in dimension hierarchies.

Named pipes. An API used for special node-to-node applications and particularly for access to communications and database servers.

Online analytical processing (OLAP). A multidimensional, multi-user, client/server computing environment for users who need to analyze consolidated enterprise data in real time. OLAP systems feature zooming, data pivoting, complex calculations, trend analyses, and modeling.

Outline. See database outline.

RDBMS. Relational database management system. A database that can be perceived as a set of tables and manipulated in accordance with the relational model of data.

Relational attribute. A characteristic of a dimension table, represented by a column. You can run SQL statements against the data in relational attribute columns.

Relational cube. A set of data and metadata that together define a multidimensional database. A relational cube is similar to an Essbase database, but refers to the portion of an Essbase database that is stored in a relational database.

Relational database. A database that is organized and accessed according to relationships between data items. A relational database contains a collection of relational tables, views, and indexes.

Relational database parameters. Parameters that you can set in the Relational Storage Manager configuration file.

Relational storage manager. A DB2 OLAP Server component that gives the OLAP engine access to DB2 and other relational databases.

Relational Storage Manager configuration file (RSM.CFG). A DB2 OLAP Server file that contains relational database parameters that you can change.

Relational table. A table that the DB2 OLAP Server creates in your relational database. The DB2 OLAP Server creates several relational tables for each Essbase application and database you create.

Relational view. A view that the DB2 OLAP Server creates in your relational database. The DB2 OLAP Server creates several relational views for each Essbase application and database you create.

Report script. An ASCII file that contains Report Writer commands that generate one or more production reports. Report scripts can be run in batch mode, using the ESSCMD command-line interface, or through the Application Manager. The script is a text file that contains data retrieval, formatting, and output instructions.

Restructure. An operation to regenerate or rebuild the tables and views that the DB2 OLAP Server created in your relational database.

RSM.CFG file. See Relational Storage Manager configuration file.

Shadow. Information stored in relational tables that shadows the information stored in an Essbase database outline.

Shared member. A member that explicitly shares storage space with another member of the same name. This member has an attribute that designates it as shared. Shared members prevent making extra calculations on a member that appears in more than one location in a database outline.

Sibling. A child member at the same branch level.

Sparse dimension. A dimension with a low percentage of available data positions filled. For example, a product that is not sold in all of a company's available markets would be a good sparse candidate.

SQL application. An application that uses SQL statements. You can use SQL applications to access data in a relational cube.

SQL. Structured Query Language. A standardized language for defining and manipulating data in a relational database.

Star schema. The type of relational database schema used by the DB2 OLAP Server. When you use the Essbase Application Manager to create an Essbase database, the DB2 OLAP Server creates a main fact table and a set of dimension tables. The fact table holds the actual data values for the database, and the dimension tables hold data about members and their relationships.

Star view. A relational view that the DB2 OLAP Server creates in your relational database that allows an SQL user to access data from the star schema in a single view with the JOIN already done.

Table space. An abstraction of a collection of containers into which database objects are stored. A table space provides a level of indirection between a database and the tables stored within the database. A table space has space on media storage devices assigned to it. Or it has tables created within it.

Time dimension. A dimension type that defines how often you collect and update data. You can tag only one dimension as Time, although you do not need to have a Time dimension.

User-defined attribute (UDA). A string saved with the member of a dimension that describes some aspect of the member. A user-defined attribute called RELANCHOR is used on the top level member of a dimension to indicate that the dimension should be used as the anchor dimension.

User-defined attribute table. A relational table that the DB2 OLAP Server creates in your relational database that contains a member ID and user-defined attribute name for each named member specified when you create an outline. There is one user-defined attribute table for each dimension in an outline.

User-defined attribute view. A relational view that the DB2 OLAP Server creates in your relational database that allows an SQL user to access all user-defined attributes for a dimension.

List of abbreviations

ADSM	ADSTAR distributed storage manager
API	Application programming interface
CLI	Call level interface
DDL	Data definition language
DRDA	Distributed relational database architecture
EIS	Executive information system
ERP	Enterprise resource planning
EWG	Essbase Web Gateway
HFS	Hierarchical file system
HIS	Hyperion Integration Server
HOLAP	Hybrid online analytical processing
IBM	International Business Machines Corporation
ITSO	International Technical Support Organization
LPAR	Logical partition
MDAT	Multidimensional analysis tools
MDSM	Multidimensional storage manager
MOLAP	Multidimensional online analytical processing
NLS	National language support
ODBC	Open database connectivity
OLAP	Online analytical processing
PSP	Preventive Service Planning
QMF	Query management facility
RACF	Resource access control facility
RDBMS	Relational database management system
ROLAP	Relational online analytical processing
RRS	OS/390 resource recovery service
RSM	Relational storage manager
SQL	Structured query language
SQLJ	Structured query language in Java
SMP	Symmetric multiprocessing
VBA	Visual Basic for application
WLM	Workload Manager
WWW	World Wide Web

Index

Symbols

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

IBM Redbooks review

Your feedback is valued by the Redbook authors. In particular we are interested in situations where a Redbook "made the difference" in a task or problem you encountered. Using one of the following methods, **please review the Redbook, addressing value, subject matter, structure, depth and quality as appropriate.**

- Use the online **Contact us** review redbook form found at http://www.redbooks.ibm.com/
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Document Number **Redbook Title**	SG24-5665-00 Getting Started with DB2 OLAP Server for OS/390
Review	
What other subjects would you like to see IBM Redbooks address?	
Please rate your overall satisfaction:	O Very Good O Good O Average O Poor
Please identify yourself as belonging to one of the following groups:	O Customer O Business Partner O Solution Developer O IBM, Lotus or Tivoli Employee O None of the above
Your email address: The data you provide here may be used to provide you with information from IBM or our business partners about our products, services or activities.	O Please do not use the information collected here for future marketing or promotional contacts or other communications beyond the scope of this transaction.
Questions about IBM's privacy policy?	The following link explains how we protect your personal information. http://www.ibm.com/privacy/yourprivacy/

SG24-5665-00
Printed in the U.S.A.

Getting Started with DB2 OLAP Server for OS/390

SG24-5665-00