

***SFA Modernization Partner***

**United States Department of Education**

**Student Financial Assistance**



**Integrated Technical Architecture  
Detailed Design Document**

**Volume 4 – Data Warehouse Architecture**

***Task Order #16***

***Deliverable # 16.1.2***

**October 13, 2000**

## Table of Contents

<b>1</b>	<b>INTRODUCTION.....</b>	<b>1</b>
1.1.	PURPOSE.....	1
1.2.	SCOPE .....	1
1.3.	APPROACH .....	1
<b>2</b>	<b>DATA WAREHOUSE ARCHITECTURE OVERVIEW.....</b>	<b>2</b>
2.1.	DATA WAREHOUSE ARCHITECTURE DOMAIN .....	2
2.2.	DATA WAREHOUSE PROCESS FLOW .....	2
2.3.	DATA WAREHOUSE PROCESS FLOW DETAILS.....	4
2.4.	DEVELOPMENT AND TEST ENVIRONMENT - HARDWARE AND INTERFACES .....	5
2.4.1.	<i>Source Systems</i> .....	6
2.4.2.	<i>Servers</i> .....	6
2.4.3.	<i>Repositories</i> .....	8
2.4.4.	<i>Data Warehouse</i> .....	8
2.4.5.	<i>Clients</i> .....	8
2.5.	DEVELOPMENT & TEST ENVIRONMENT SPECIFICATIONS .....	9
2.6.	INTERFACES AND ADAPTORS OVERVIEW .....	11
2.7.	DATA WAREHOUSE ARCHITECTURE REQUIREMENTS AND GUIDING PRINCIPALS .....	12
2.7.1.	<i>Scalability</i> .....	12
2.7.2.	<i>Reliability</i> .....	12
2.7.3.	<i>Flexibility</i> .....	13
2.7.4.	<i>Performance</i> .....	13
2.7.5.	<i>Security</i> .....	13
2.8.	PRODUCTION ENVIRONMENT .....	13
<b>3</b>	<b>INFORMATICA OVERVIEW.....</b>	<b>14</b>
3.1.	INTRODUCTION .....	14
3.2.	POWERCENTER .....	14
3.2.1.	<i>Client Workstation</i> .....	14
3.3.	POWERCENTER DESIGNER MODULE.....	14
3.3.1.	<i>Source Analyzer</i> .....	16
3.3.2.	<i>Warehouse Designer</i> .....	16
3.3.3.	<i>Transformation Developer</i> .....	17
3.4.	POWERCENTER SERVER MANAGER MODULE.....	17
3.5.	POWERCENTER REPOSITORY MANAGER MODULE.....	18
<b>4</b>	<b>RDBMS OVERVIEW .....</b>	<b>21</b>
4.1.	DESIGN ISSUES.....	21
4.2.	SIZING AND CONFIGURATION .....	21
4.3.	DATA MODEL.....	21

---

4.4.	BACK-UP AND RECOVERY.....	22
4.5.	DATA LOAD .....	22
<b>5</b>	<b>MICROSTRATEGY OVERVIEW .....</b>	<b>23</b>
5.1.	INTRODUCTION .....	23
5.2.	MICROSTRATEGY INTELLIGENCE SERVER .....	23
5.3.	MICROSTRATEGY METADATA REPOSITORY.....	23
5.4.	MICROSTRATEGY DESKTOP .....	23
5.4.1.	<i>MicroStrategy Architect</i> .....	24
5.4.2.	<i>MicroStrategy Agent</i> .....	24
5.4.3.	<i>MicroStrategy Administrator</i> .....	24
5.5.	MICROSTRATEGY WEB.....	25
5.6.	MICROSTRATEGY BROADCASTER.....	25
5.7.	MICROSTRATEGY INFOCENTER .....	25
<b>6</b>	<b>DATA WAREHOUSE SERVICES .....</b>	<b>26</b>
6.1.	METADATA LOOKUP .....	26
6.2.	ROLAP (CLIENT / SERVER).....	26
6.3.	ROLAP (WEB) .....	26
6.4.	DSS ADMINISTRATION .....	27
6.5.	EXTRACTING .....	27
6.6.	TRANSFORMING.....	27
6.7.	LOADING AND INDEXING .....	27
6.8.	QUALITY ASSURANCE CHECKING.....	27
6.9.	RELEASE/PUBLISHING .....	28
6.10.	UPDATING .....	28
6.11.	QUERYING.....	28
6.12.	DATA FEEDBACK .....	28
6.13.	AUDITING .....	28
6.14.	SECURING.....	28
6.15.	BACKING UP AND RECOVERING.....	28
<b>7</b>	<b>DATA WAREHOUSE NAMING CONVENTIONS.....</b>	<b>30</b>
7.1.	INTRODUCTION INFORMATICA POWERCENTER STANDARDS.....	30
7.2.	MICROSTRATEGY NAMING CONVENTIONS .....	30
7.3.	INFORMATICA DIRECTORY STRUCTURES .....	31
7.3.1.	<i>PowerCenter Server</i> .....	31
7.3.2.	<i>PowerCenter Client</i> .....	31
7.4.	MICROSTRATEGY DIRECTORY STRUCTURES .....	31
7.4.1.	<i>MicroStrategy Intelligence Server 7.0</i> .....	32
7.4.2.	<i>MicroStrategy Web 7.0</i> .....	32
7.4.3.	<i>MicroStrategy Desktop 7.0</i> .....	32

---

<b>8</b>	<b>DATA WAREHOUSE APPLICATION DESIGN.....</b>	<b>33</b>
8.1.	INTRODUCTION .....	33
8.2.	REPORT INTERFACES FOR USERS.....	33
8.2.1.	<i>ROLAP (Web)</i> .....	33
8.2.2.	<i>ROLAP (Client / Server)</i> .....	33
<b>9</b>	<b>DATA WAREHOUSE PROGRAMMING CHOICES.....</b>	<b>34</b>
9.1.	INTRODUCTION .....	34
9.2.	INFORMATICA .....	34
9.2.1.	<i>External Procedure Transformations</i> .....	34
9.2.2.	<i>Informatica PowerCenter.e</i> .....	34
9.3.	MICROSTRATEGY PROGRAMMING CHOICES.....	35
9.3.1.	<i>MicroStrategy Intelligence Server, MicroStrategy Web, and MicroStrategy Desktop 7.0..</i> .....	35
<b>10</b>	<b>DATA WAREHOUSE CONFIGURATION.....</b>	<b>36</b>
10.1.	INTRODUCTION .....	36
10.2.	POWERCENTER CLIENT WORKSTATION .....	37
10.2.1.	<i>Installation Prerequisites</i> .....	37
10.2.2.	<i>Installation and Configuration Guidelines</i> .....	37
10.3.	POWERCENTER SERVER.....	38
10.3.1.	<i>Installation Prerequisites</i> .....	38
10.4.	POWERCENTER REPOSITORY.....	39
10.4.1.	<i>Prerequisites</i> .....	39
10.5.	MICROSTRATEGY DESKTOP (MICROSTRATEGY AGENT, MICROSTRATEGY ARCHITECT, MICROSTRATEGY ADMINISTRATOR) 7.0.....	40
10.5.1.	<i>Prerequisites</i> .....	40
10.5.2.	<i>Installation Prerequisites</i> .....	40
10.5.3.	<i>Installation and Configuration Guidelines</i> .....	41
10.6.	MICROSTRATEGY INTELLIGENCE SERVER 7.0 .....	42
10.6.1.	<i>Installation Prerequisites</i> .....	42
10.6.2.	<i>Installation and Configuration Guidelines</i> .....	42
10.7.	MICROSTRATEGY WEB 7.0.....	43
10.7.1.	<i>Installation Prerequisites</i> .....	43
10.7.2.	<i>Installation and Configuration Guidelines</i> .....	43
10.8.	METADATA REPOSITORY .....	45
10.9.	MICROSTRATEGY SERVER HARDWARE SIZING .....	46
10.9.1.	<i>MicroStrategy Web and Intelligence Server 7.0 Sizing</i> .....	46
10.9.2.	<i>Metadata Repository Server Sizing</i> .....	49
<b>11</b>	<b>DATA WAREHOUSE SECURITY ARCHITECTURE .....</b>	<b>50</b>
11.1.	INTRODUCTION .....	50

---

11.2.	INFORMATICA SECURITY .....	50
11.2.1.	<i>Introduction</i> .....	50
11.2.2.	<i>Securing Development Environments</i> .....	50
11.2.3.	<i>Securing Production Environments</i> .....	51
11.2.4.	<i>User Groups</i> .....	51
11.2.5.	<i>Editing a User Password</i> .....	54
11.2.6.	<i>Repository Privileges</i> .....	54
11.2.7.	<i>Locking Within Objects</i> .....	55
11.2.8.	<i>Locking with Cubes and Dimensions</i> .....	55
11.2.9.	<i>Locking Business Components</i> .....	56
11.2.10.	<i>Handling Locks</i> .....	56
11.2.11.	<i>Viewing a Lock</i> .....	56
11.2.12.	<i>Tips</i> .....	56
11.2.13.	<i>Create groups with limited privileges</i> .....	57
11.3.	DATA WAREHOUSE RDBMS.....	58
11.3.1.	<i>Security Views</i> .....	58
11.3.2.	<i>Partitioned fact tables</i> .....	58
11.3.3.	<i>Split fact tables</i> .....	58
11.4.	MICROSTRATEGY SECURITY FEATURES.....	59
11.4.1.	<i>MicroStrategy Application Security</i> .....	59
11.4.2.	<i>MicroStrategy 7.0 Access Control</i> .....	60
<b>12</b>	<b>DATA WAREHOUSE PERFORMANCE CONSIDERATIONS.....</b>	<b>64</b>
12.1.	INTRODUCTION .....	64
12.2.	DATABASE TUNING TECHNIQUES .....	64
12.2.1.	<i>Aggregate tables</i> .....	64
12.2.2.	<i>Indices</i> .....	65
12.2.3.	<i>Denormalization</i> .....	65
12.2.4.	<i>Partitioning</i> .....	65
12.2.5.	<i>Preliminary Database Tuning Methodology</i> .....	66
12.3.	MICROSTRATEGY INTELLIGENCE SERVER THREAD MANAGEMENT .....	67
12.4.	INFORMATICA POWERCENTER .....	67
<b>13</b>	<b>DATA WAREHOUSE OPERATIONS ARCHITECTURE.....</b>	<b>71</b>
13.1.	INTRODUCTION .....	71
13.2.	INFORMATICA .....	71
13.3.	DATA WAREHOUSE .....	71
13.4.	MICROSTRATEGY .....	71
<b>14</b>	<b>DATA WAREHOUSE ADMINISTRATIVE SPECIFICATIONS.....</b>	<b>72</b>
14.1.	SUN SOLARIS ADMINISTRATION PROCEDURES .....	72
14.1.1.	<i>Informatica PowerCenter Server Start-up Procedures</i> .....	72

14.1.2.	<i>Informatica PowerCenter Server Shutdown Procedures</i>	72
14.1.3.	<i>Sun Solaris Backup Procedures</i>	72
14.1.4.	<i>Sun Solaris Recovery Procedures</i>	73
14.1.5.	<i>Sun Solaris Performance and Tuning</i>	73
14.2.	NT ADMINISTRATION PROCEDURES	74
14.2.1.	<i>General Startup Procedures for MicroStrategy products</i>	74
14.2.2.	<i>MicroStrategy Intelligence Server Startup Procedures</i>	74
14.2.3.	<i>MicroStrategy Intelligence Server Shut down Procedures</i>	75
14.2.4.	<i>MicroStrategy Intelligence Server Backup</i>	75
14.2.5.	<i>General NT Performance and Tuning Recommendations</i>	75
<b>15</b>	<b>GLOSSARY OF TERMS AND ACRONYMS</b>	<b>77</b>
15.1.	TERMS	77
15.2.	ACRONYMS	80
<b>APPENDIX A</b>		<b>82</b>

### **List of Figures**

Figure 1 – SFA Technical Architecture Domains.....	2
Figure 2 – Data Warehouse Process Flow .....	3
Figure 3 – Data Warehouse Process Flow Details .....	4
Figure 4 – Development and Test Environment – Hardware and Interfaces.....	6
Figure 5 - Mapping Designer Window - Open .....	15
Figure 6 – Server Manager Module Window - Open .....	18
Figure 7 – Repository Manager Module - Open .....	19
Figure 8 – MicroStrategy Desktop - Open.....	24

### **List of Tables**

Table 1 – Development & Test Environment Specifications .....	9
Table 2 – Data Warehouse Component Communication Protocols.....	11
Table 3 – Informatica PowerCenter Transformation Standards.....	30
Table 4 – Directory Structure – PowerCenter Server: Unix .....	31
Table 5 – Directory Structure - MicroStrategy: Windows.....	31
Table 6 –Directory Structure – MicroStrategy Intelligence Server 7.0.....	32
Table 7 – Directory Structure – MicroStrategy Web 7.0.....	32
Table 8 – Directory Structure – MicroStrategy Desktop 7.0.....	32
Table 9 – Application Scenario Concurrent Users to Total Users Ratios .....	47
Table 10 – Potential Hardware Configurations.....	48
Table 11 – Recommended Configuration Based on Number of Concurrent Users And Desired Response Time .....	49
Table 12 – Repository Locks.....	55
Table 13 – List of Terms.....	77
Table 14 – List of Acronyms.....	80

## 1 Introduction

### 1.1 Purpose

The purpose of this document is to provide the detailed design specifications necessary to build and maintain the Data Warehouse Architecture (DWA) for the Student Financial Assistance (SFA) Department.

### 1.2 Scope

Building a data warehouse is an iterative process ultimately driven by project requirements. As such, this document will provide the design details needed to implement the DWA within the context of the Chief Financial Office (CFO) Datamart and Central Data System (CDS) Retirement projects, which are the two SFA data warehouse projects currently underway. However, the infrastructure recommended here is designed not only to support these current business requirements, but also to be fully scalable as the organizations' data warehousing needs expand and grow over time.

In addition, the design specifications provided here take into account the actual hardware currently available. Where the actual design deviates materially from an optimal configuration, this deviation will be noted and explained in detail.

### 1.3 Approach

This document begins with a discussion of the overall DWA and then addresses in detail the individual applications, their services, and their interfaces.

## 2 Data Warehouse Architecture Overview

### 2.1 Data Warehouse Architecture Domain

For purposes of this document, a data warehouse is defined as a subject oriented, integrated, time-variant, non-volatile collection of data used to support the decision making process of an organization. Data in the data warehouse is generally a copy of transaction data (though non-transaction data may also be included) and is specifically structured for querying and reporting. The data warehouse will reside on disks and servers that are separate from the SFA’s transaction processing systems and be configured specifically to facilitate speedy and accurate querying and reporting.

Figure 1 shows the five separate domains of the SFA Technical Architecture and represents the Data Warehouse as a subset of the overall architecture.

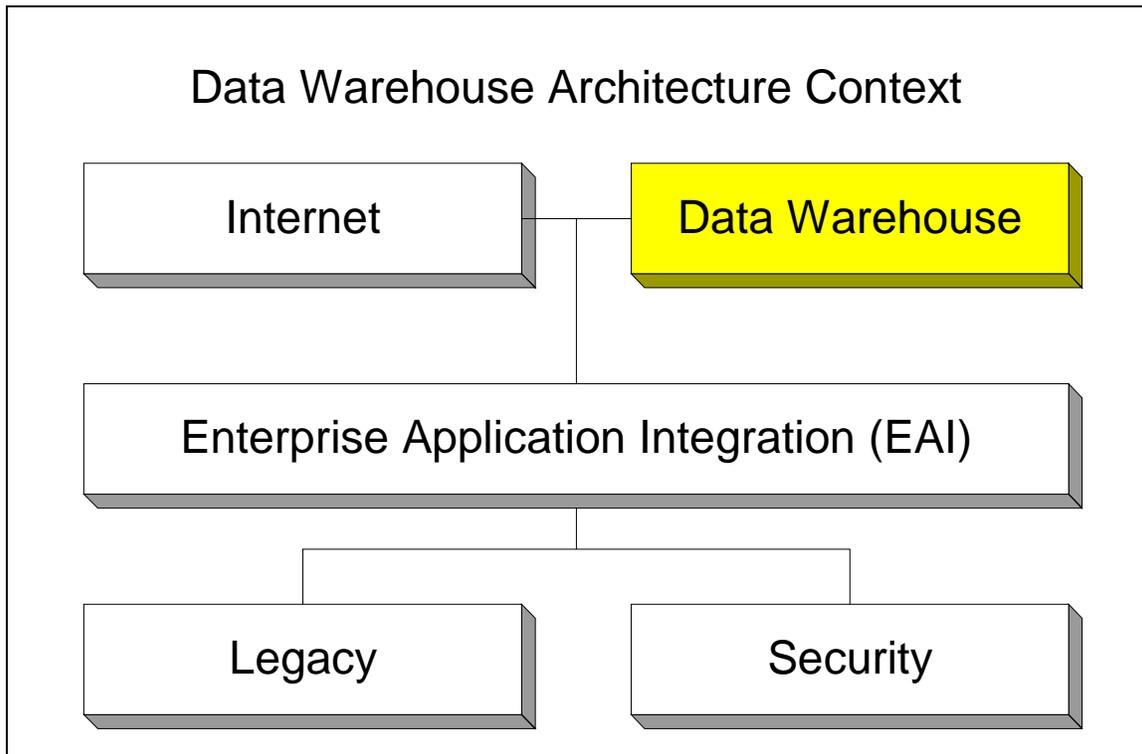


Figure 1 – SFA Technical Architecture Domains

### 2.2 Data Warehouse Process Flow

In this section we discuss the process flow between the various applications or “tools” that comprise the Data Warehouse. In the following sections we will build on this discussion and

address the more technical aspects of these processes. Figure 2 provides a high-level illustration of this process flow.

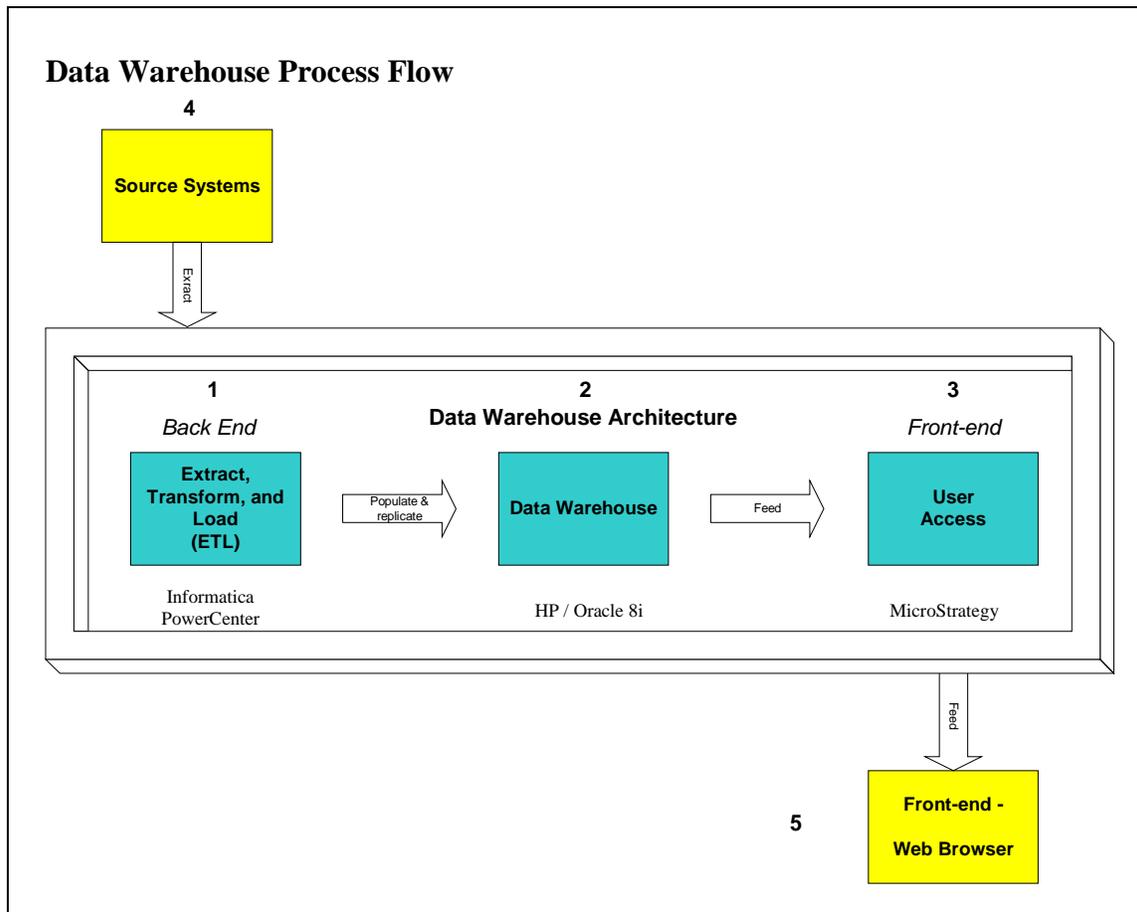


Figure 2 – Data Warehouse Process Flow

The DWA (Figure 2) is comprised of the following three components:

1. *The Extraction, Transformation, and Loading Process (ETL)*, also known as the “back-end.” This is provided by Informatica’s PowerCenter 1.7.
2. The *Data Warehouse*, which resides on an Oracle 8i Relational Database Management Systems (RDBMS).
3. *User Access*, also known as the “front-end” is provided by MicroStrategy 7.0 Platform and which enables the user interactive reporting capabilities such as drilling, pivoting, and report creation.

The following components reside outside the DWA:

4. *Source systems*, which are the operational system of record whose function is to capture the transactions. In the SFA environment these will primarily consist of legacy systems including DB2 mainframe data, Informix databases, and flat files.

- The *Web Browser* is provided by MicroStrategy Web and enables the user the same interactive reporting capabilities through a web browser rather than the client application.

### 2.3. Data Warehouse Process Flow Details

Figure 3 “drills down” the DWA into a detail to provide greater detail. In the following section we elaborate more on the various DWA processes, their specific components, and also introduce the concept of metadata into the process flow.

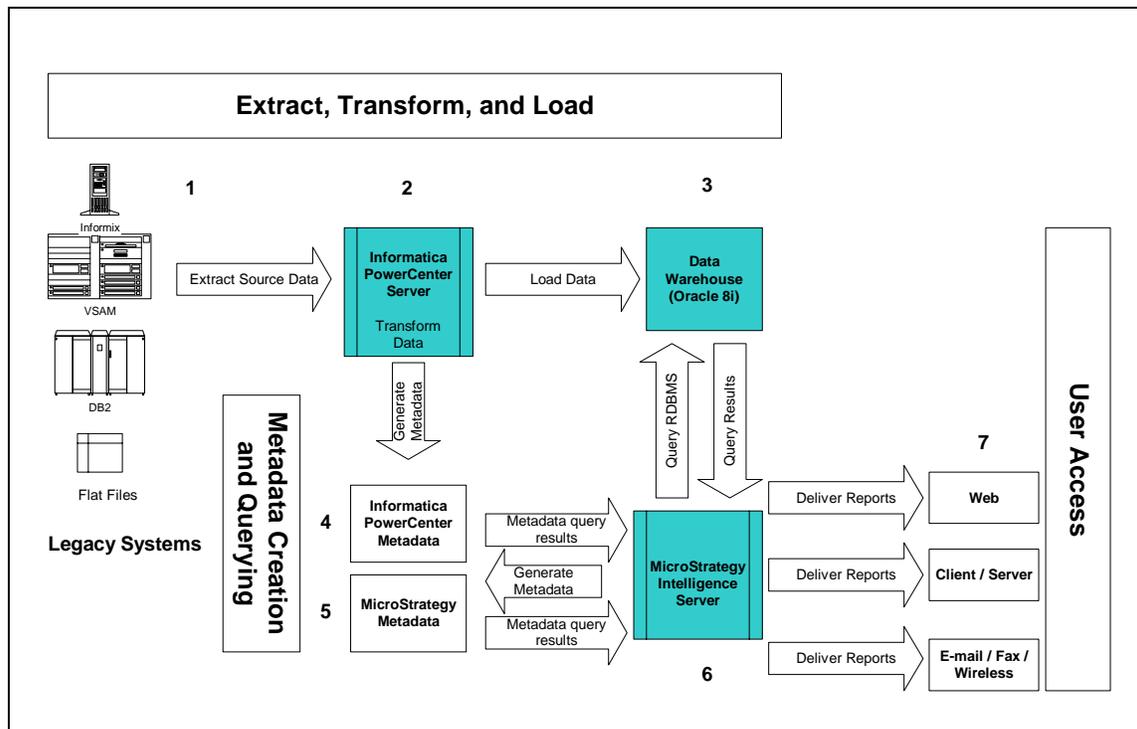


Figure 3 – Data Warehouse Process Flow Details

**Extract, Transform, and Load** - The ETL process consists of several steps. First is the extraction or reading the source data. Once data is extracted, numerous transformation steps may be undertaken, including cleansing erroneous data, purging unnecessary fields, combining sources, and building aggregates. Loading, the final step in the ETL process, is the activity of replicating dimension and fact tables and presenting them to the data warehouse via bulk loading facilities.

**User Access** - Once data is loaded into the Data Warehouse, User Access can occur. In the User Access module, the MicroStrategy server processes query requests initiated by the front-end web browser, translates these requests into SQL statements and returns results by delivering reports to the user via the web-browser.

**Metadata Creation and Querying** – Metadata is created during both the ETL and User Access process. Both the Informatica and MicroStrategy applications create metadata that is stored in their respective repositories.

- 1) **Legacy Systems** – These will provide the primary source data and will include VSAM files, DB2 Mainframe data, and flat files.
- 2) **Informatica PowerCenter Server** – is the actual engine that provides the data extraction, transformation, population services as well as creation and management of Informatica Metadata.
- 3) **Data Warehouse** – the Oracle 8i RDBMS where the data physically resides.
- 4) **Informatica PowerCenter Metadata** – will provide the technical metadata regarding data sources, field names, target tables, etc. primarily aimed at developers and Database Administrator (DBA) resources.
- 5) **MicroStrategy Metadata** – Provides the business metadata, generally aimed at the business users seeking file structure definitions, business rules.
- 6) **MicroStrategy Intelligence Server** - is the middle-tier between the user applications and the data warehouse providing report cache management, analytical functions, job prioritization, and thread management.
- 7) **Web / Client Server / E-mail / Fax / Wireless** – the various outputs available for reports created by the MicroStrategy 7 Platform.

## **2.4 Development and Test Environment - Hardware and Interfaces**

Figure 4 provides a detailed view of the hardware, repositories and communication protocols for the DWA Development and Test environment installed at the Virtual Data Center (VDC). Following the diagram we provide an overview of the individual components and elaborate on their functionality.

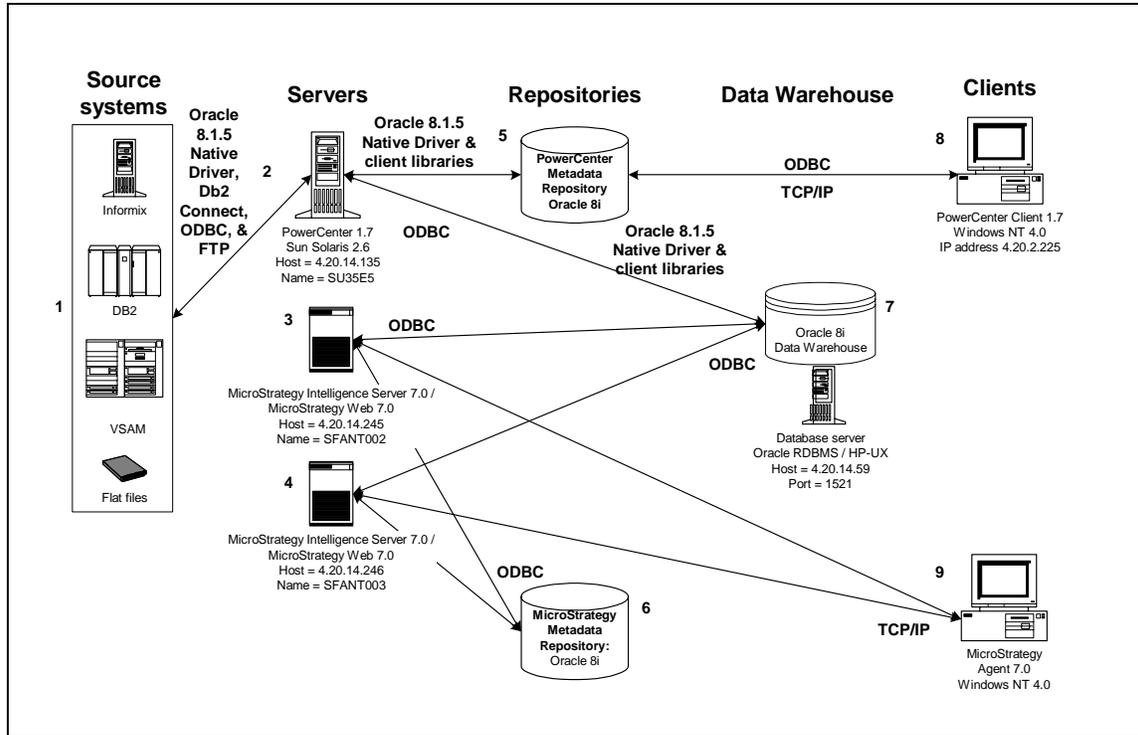


Figure 4 – Development and Test Environment – Hardware and Interfaces

### 2.4.1. Source Systems

- 1) **Source Systems** - These are the “legacy” systems that will be the primary source of data for the warehouse. For the CDS Retirement project the source data will come from flat files transferred to the Informatica server via File Transfer Protocol (FTP). The CFO project will use data from the Financial Management System (FMS) that resides in an Oracle 8.0.5 database.

### 2.4.2. Servers

- 2) **PowerCenter 1.7 Server** - The server is used to read, transform, and write data between sources and targets. The Informatica Server moves data from sources to targets based on metadata stored in a repository and also performs the following tasks:

- Manages the scheduling and execution of sessions and batches
- Executes sessions and batches
- Verifies permissions and privileges
- Interacts with the Server Manager and the supplied command line program.

The PowerCenter server uses the following protocols to connect with sources, targets, and repositories:

**ODBC and Native Drivers** - In the development environment shown above, the PowerCenter server communicates with the Oracle 8i repository and data warehouse via the native drivers and client libraries installed on it. Native drivers will also connect the server to Informix data sources; DB2 Connect (to be installed on the server) will provide communication between the server and the DB2 mainframe, while Open Database Connectivity (ODBC) will connect the server to all other data sources.

**FTP** - The Informatica Server can also use the File Transfer Protocol (FTP) to access source and target files. With both source and target files, files can be FTP'd directly to the Informatica Server or staged on a local directory for the session run. Files can also be staged by creating a pre-session shell command to move the files local to the Informatica Server. However, accessing files directly with FTP generally provides better session performance than using FTP to stage the files. When using FTP file sources and targets in a session, the following is needed:

- FTP connection name
- Remote file name and exact path
- Whether you want to stage the files

*Mainframe Note* - Due to mainframe restrictions, the following constraints apply when using this feature with mainframe machines:

- While FTP can be used to source files from mainframe machines, it is not possible to FTP target files to mainframes.
- It is not possible to run sessions concurrently (in a batch or as individual sessions) if the sessions use the same FTP source file located on a mainframe.
- If a batch is aborted that contains a session with a staged FTP source origination from a mainframe, a connection to timeout must occur before the batch or session can run again.

3) **MicroStrategy Intelligence Server 7.0, MicroStrategy Web 7.0** – MicroStrategy Web provides On-Line Analytical Processing (OLAP) over the Internet through a web browser. It provides a customizable Hypertext Markup Language (HTML) interface. MicroStrategy Web provides the following services:

- Viewing or grid and graph report
- Report creation
- Report drilling and pivoting

MicroStrategy Web requires Microsoft Internet Information Server to provide web-hosting functionality. Web users will communicate with MicroStrategy Web via Hypertext Transfer Protocol (HTTP) requests.

MicroStrategy Intelligence Server acts as a middle-tier between MicroStrategy Web and the data warehouse. It provides caching, thread management, and scheduling capabilities.

- 4) **Fail Over Server** – MicroStrategy Intelligence Server 7.0 and MicroStrategy Web 7.0 will be installed on this fail over server to provide application availability if Machine 3 stops functioning. The MicroStrategy 7 product line provides built-in fail over support but will require the International Business Machine (IBM) eNetwork Dispatcher to transfer requests via Internet Protocol (IP) between the primary web application server (Machine 3) and the fail over server (Machine 4). The products on this machine communicate with the databases and repositories as described above.

### 2.4.3. Repositories

- 5) **PowerCenter Metadata Repository** - contains metadata the server uses to transform data from sources to targets. Informatica Metadata Exchange (MX) provides a set of relational views that allow easy Structures Query Language (SQL) access to the Informatica metadata repository. MX views provide information to analyze metadata stored in the repository including:
- Database definition metadata
  - Source metadata
  - Target metadata
  - Session metadata
  - Mapping and transformation metadata
  - Star and multi-dimensional schema metadata
- 6) **MicroStrategy Metadata repository** - The metadata repository contains detailed information on the data warehouse tables, attributes, facts, and relationships. All information for a MicroStrategy Project is stored in the metadata repository. All reporting objects including templates, filters, and reports for all users are also stored in the metadata repository. This central metadata repository architecture allows for reports to be created once and deployed through any of the MicroStrategy applications. The metadata repository will be stored in an Oracle database on the same machine as the data warehouse.

### 2.4.4. Data Warehouse

- 7) **Oracle 8i Data Warehouse** – is the server that provides the Relational Database Management System (RDBMS) services for the data warehouse, the data warehouse tables, as well as related objects (triggers, procedures, etc.) and the tables for the Informatica repository. As the data warehouse grows tables will be stored on a Storage Area Network (SAN).

### 2.4.5. Clients

- 8) **PowerCenter Client** – is a Windows NT workstation where development, administration, and operations will be performed for the Informatica Extract Transform and Load (ETL) jobs. PowerCenter 1.7 client tools use ODBC drivers to connect to source

and target databases. PowerCenter supports the Merant 32-bit ODBC drivers included on the Informatica installation Compact Disk (CD) for Informix, Oracle, and Sybase databases. For DB2, Microsoft SQL Server, Microsoft Excel, and Microsoft Access97 databases, a vendor-supplied ODBC driver is needed.

- 9) **MicroStrategy Desktop 7.0** - provides OLAP functionality including report drilling and pivoting along with statistical, financial, and OLAP functions. Drilling is the process of requesting additional data based on a different attribute of aggregation level. Pivoting in a report grid is the process of rotating the rows and columns to see different summaries of the source data. Developers and power users can use this tool to create reports, define business metrics, and perform data analysis.

## 2.5. Development & Test Environment Specifications

The following table lists the servers, hardware specifications, applications, and software specifications required for the data warehouse development and test environment. The Machine Purpose column lists the functionality that the machine will provide. Hardware Specifications lists the central processing unit (CPU), random-access memory (RAM), and hard drive specification on each server machine. The Primary Application column lists the Informatica or MicroStrategy application that will be running on the server machine. Software Requirements lists other software applications required for the primary application, as well as any prerequisites.

*Note that the hardware and software specifications that follow do not represent an optimal design, but instead show the actual configuration that exists at the VDC which was determined using the equipment that was available.*

Table 1 – Development & Test Environment Specifications

#	Machine Purpose	Hardware Specifications	Primary Application	Software Requirements
1	Source Systems	Existing legacy systems	DB2, Informix, Flat Files	N/A
2	Informatica Server	Sun Enterprise 3500 4 Processors/CPU's 366 Mhz 4 GB of RAM (1 GB per processor) 8 System Boards 10 GB Disk Space	PowerCenter Server	Sun Solaris 2.6 or 2.7 TCP/IP network protocol 32 bit database client utilities installed for repository, all sources and targets Client can access Oracle via Net 8 and sql plus; Informix via ESQ/L/C and ISQL, DB2 via DB2 connect and Dbaccess, etc.

#	Machine Purpose	Hardware Specifications	Primary Application	Software Requirements
3	MicroStrategy Intelligence Server 7.0, Web 7.0	Dual Pent III 800 MHz 1 GB RAM 48 GB Disk Space	MicroStrategy Intelligence Server 7.0, MicroStrategy Web 7.0	Windows NT 4.0 SP5 MS Internet Information Server 4.0 3 MB of memory for registry MS Internet Explorer 5.0
4	<b>MicroStrategy Failover:</b> MicroStrategy Intelligence Server 7.0, Web 7.	Dual Processor Pentium PIII, 800 MHz 1 GB RAM 48 GB disk space	MicroStrategy Intelligence Server 7.0, MicroStrategy Web 7.0	Windows NT 4.0 SP5 MS Internet Information Server 4.0 3 MB of memory for registry MS Internet Explorer 5.0
5	PowerCenter Metadata Repository	See machine #7 below	Oracle	See machine #7 below
6	MicroStrategy Metadata Repository	See machine #7 below	Oracle	See machine #7 below
7	Oracle 8i Data Warehouse	HP V Class server 16 processors, 552Mhz 16GB RAM 4 x 18GB SCSI Disk	HP UX 11.0	Oracle 8i

#	Machine Purpose	Hardware Specifications	Primary Application	Software Requirements
8	PowerCenter Client 1.7	64 MB RAM (Minimum requirement) 40 MB Disk space	PowerCenter client	Windows 95/98 or NT  the windows temp variable must point to a valid directory  the login must have administrator rights on the client  32 bit database client utilities installed for repository, all sources and targets  Client can access Oracle via Net 8 and sql plus; Informix via ESQ/L/C and ISQL, DB2 via DB2 connect and Dbaccess, etc.  environment or regional setting (code page) compatible with server and repository  TCP/IP protocol configured to communicate with the PowerCenter Server host
9	MicroStrategy Desktop 7.0	Pentium, 266 MHz 64 MB RAM (Minimum requirement) 2 GB disk space (40 MB for installation)	MicroStrategy Agent 7.0, MicroStrategy Architect 7.0, MicroStrategy Administrator 7.0	Windows 95 DCOM (installs with product) IE 4.01 SP1

## 2.6 Interfaces and Adaptors Overview

The Data Warehouse will be interfacing with several systems. Some systems are located in the Internet space while others are legacy systems located in the back-end. The interfaces between the Data Warehouse and other systems may require adapters or interfaces.

The table below lists the communication protocols used to communicate between each of the data warehouse components:

Table 2 – Data Warehouse Component Communication Protocols

	Machine	Protocol	Notes
1	Source Systems	ODBC or Native Drivers	Oracle and Informix will use native drivers; DB2 will access via DB2 Connect; other sources will be accessed via ftp or odbc
2	Informatica Server	ODBC or Native Drivers	Oracle and Informix will use native drivers; DB2 will access via DB2 Connect; other sources will be accessed via ftp or odbc

	<b>Machine</b>	<b>Protocol</b>	<b>Notes</b>
3	MicroStrategy Intelligence Server 7.0, Web 7.0, InfoCenter 6.5	TCP/IP / ODBC	
4	MicroStrategy Failover: MicroStrategy Intelligence Server 7.0, Web 7.	ODBC	
5	PowerCenter Metadata Repository	Native drivers	
6	MicroStrategy Metadata Repository	ODBC	
7	Oracle 8i Data Warehouse	Native drivers	
8	PowerCenter Client 1.7	TCP/IP / ODBC	
9	MicroStrategy Desktop 7.0	ODBC	

## 2.7. Data Warehouse Architecture Requirements and Guiding Principals

The DWA adheres to a number of specific design principals, which are summarized below.

### 2.7.1. Scalability

Scalability is the ability to increase capacity as demands increase, as data stores grow, as more users are added to the system, and as more applications are developed against the Warehouse. Often as the users become familiar with the query capabilities their success will cause them to demand more from the system. In time, user queries become more sophisticated and reporting needs become more complex. All of these are factors that demand scalability in a system. The proposed DWA provides substantial flexibility and scalability by allowing changes and enhancements to occur within an individual piece of the architecture without having a major effect on the entire system. For example, if additional capacity is needed to accommodate larger volumes of data, the RDBMS can be expanded (by adding additional storage) while the other two components (ETL and User Access) can remain unchanged. If additional users require access to reporting, then the User Access component can be expanded. This ability to expand components provides true scalability for the entire DWA.

### 2.7.2. Reliability

The Data Warehouse must provide a high level of reliability. Software and hardware components within the architecture provide the required level of reliability by utilizing redundancy and fail-over technologies.

### **2.7.3. Flexibility**

The DWA must be flexible enough to utilize data from a variety of sources, populate and query virtually any RDBMS and database schema, deliver reports to virtually any medium. Furthermore, the Warehouse must be adaptable to changes in the enterprise, such as reorganizations, mergers, and acquisition as it may be necessary to implement new queries after such changes—queries not envisioned in the original design.

### **2.7.4. Performance**

Processing should be distributed over several applications and database systems to provide the best overall system performance. The design should be requirements for completing processing

### **2.7.5. Security**

Security can be implemented within each portion of the architecture and as an integrated subsystem of the overall technical architecture system. Security is a large part of each portion of the Data Warehouse, ETL, database, and relational OLAP (ROLAP).

## **2.8. Production Environment**

The production environment is not yet installed so the topology has not yet been determined.

## 3 Informatica Overview

### 3.1 Introduction

Informatica's PowerCenter provides ETL services for the DWA. The following section outlines PowerCenter's basic functionality.

### 3.2 PowerCenter

Informatica PowerCenter is a GUI-based, enterprise data integration that enables an organization to transform legacy, relational and enterprise resource planning (ERP) data into information for strategic business analysis.

Within an enterprise decision support solution, PowerCenter is responsible for extracting data from operational sources, transforming it if necessary, cataloging it for use and re-use, and delivering it to business intelligence and analytic applications. PowerCenter 1.7 is a metadata-driven product that utilizes an enterprise data integration hub, including:

- Consolidation, cleansing and customization of data
- Integration of operational and analytical resources
- Centralized management of distributed resources

#### 3.2.1 Client Workstation

The PowerCenter client utilizes three modules to perform the development, administration, and session scheduling for PowerCenter services. The modules are the Designer, Server Manager, and Repository Manager. Each of these modules is described below.

### 3.3 PowerCenter Designer Module

The Designer is comprised of four integrated modules: *Source Analyzer*, *Warehouse Designer*, *Mapping Designer* and *Transformation Developer*. This is the key tool used by Data Integrator Developers. The tool allows the developers to build the mappings, which are the cornerstone of the Extract, Transforming, and Loading (ETL) process.

Mappings typically are setup to export the data from source tables, which can be legacy or other systems, modify the data using business logic, and then loads the data into target tables, in this case Oracle tables, for later retrieval, querying, and reporting. The Designer tool is where the actual building of the ETL phase is done. The tool is loaded on the developer's workstations.

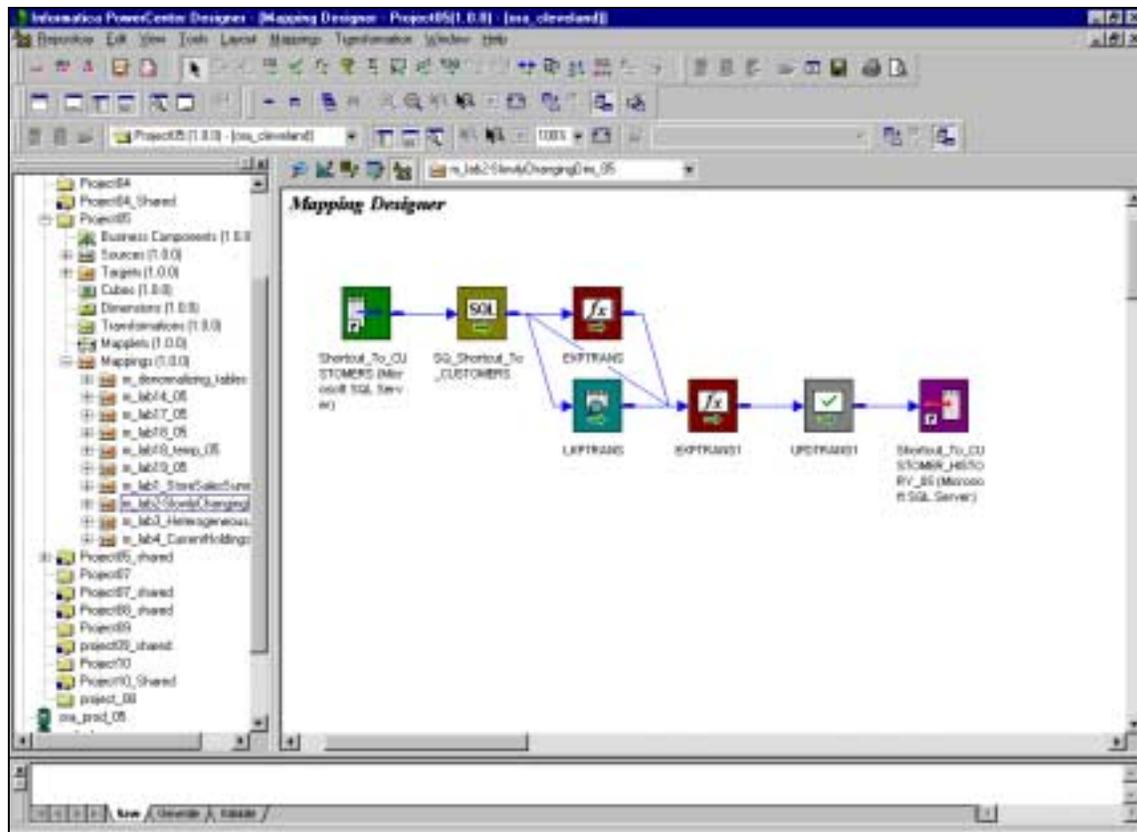


Figure 5 - Mapping Designer Window - Open

The Mapping Designer gives developers a visual aid to creating and editing source-to-target mappings. To do that, it employs an approach known as dataflow diagramming.

Dataflow diagramming is an intuitive method of creating dataflow links through combinations of PowerCenter 1.7's transformation objects. Sources, targets and transformation objects can be dragged and dropped into a workspace to construct the transformation pipeline.

The transformation part of the ETL process is accomplished using the following PowerCenter 1.7 objects in the mapping Designer:

- *Aggregator* - Performs an aggregate calculation (such as sum, average or count) on a column's worth of data;
- *Expression* -- Perform custom calculations of a simple or complex nature, using data from one or more input ports;
- *External procedure* - Calls a procedure defined in a shared library;
- *Filter* - Performs a test on all records before allowing them to be sent to the next object;
- *Joiner* - Joins data from disparate sources, such as mainframes, flat files and relational databases;
- *Lookup* - Looks up values;

- *Normalizer* - Expands VSAM files and processes COBOL "Occurs" clauses;
- *Rank* - Performs comparisons and groupings;
- *Sequence generator* - Generates unique ID values in the same fashion as a sequence in a relational database;
- *Source qualifier* - Represents data temporarily stored on the data mart server;
- *Stored procedure* - Calls a stored procedure and captures return values;
- *Update strategy* - Defines how the data mart server should handle updates to existing records in targets.

By linking together multiple objects, dataflow diagramming gives developers access to a powerful but simple mechanism for creating and modifying even the most complex transformations.

### **3.3.1. Source Analyzer**

The Source Analyzer reads, analyzes and "reverse engineers" the structures (schema information) of operational databases and flat files, then stores that information in the repository, to be used by the other PowerCenter components.

The Source Analyzer reads RDBMS native catalogs and mainframe flat-file definitions, and stores information such as table names, field names, types and sizes. When extracting operational schema from a Sybase database, for example, the Source Analyzer extracts the tables, field names and field types directly from the Sybase system catalog. Table structures, field types and table relationships (such as primary and foreign key relationships) can then be verified for accuracy before designing the data mart schema.

After extraction, equivalent structures can be edited or deleted to refine the structure of the data definition. This allows fields to be combined or rearranged so they accurately indicate their contents.

### **3.3.2. Warehouse Designer**

The Warehouse Designer is used to design and edit the schema of the data warehouse or data mart, which is composed of target tables.

PowerCenter 1.7 gives developers several options for creating the warehouse schema.

- Developers can enter target table definitions directly, or may create a target definition by first replicating and then interactively rearranging an existing source table definition.
- Developers can also employ PowerCenter's multi-dimensional dimension and cube editors, which provide an automated, wizard-based mechanism for creating a variety of multi-dimensional schema, such as star, snow-flake, constellation, and redundant.

The Dimension Editor allows the developer to create a new dimension or edit and delete an existing one. The developer can subsequently add levels and hierarchies for the newly created dimension or defer this task to a later time.

### **3.3.3. Transformation Developer**

The Transformation Developer makes possible the creation and sharing of reusable transformations.

To create a developer-defined transformation, the developer at any time can move from the Mapping Designer into the Transformation Developer module, and employ any of the PowerCenter 1.7 transformation objects (Expression, Aggregator, Lookup, and so on) as the basis for the new transformation.

Alternatively, the developer can employ another PowerCenter technology, the Transformation Expression (TX) Application Programming Interface (API), to import and register as PowerCenter transformation objects custom routines written in C, C++, Visual Basic or other popular languages.

The developer then adds a copy or an instance of the newly defined transformation to the appropriate mapping. The difference depends on the developer's intent and objectives: a copy will remain static, but an instance automatically inherits any subsequent changes made to the original developer-defined transformation.

One use for such a transformation might be as a standard filter for excluding closed records. Once the correct logic for detecting closed records has been determined and the developer-defined transformation object created, other developers within the department can simply take that object for their own mappings, thus saving time and effort, and ensuring department-wide consistency.

It should be noted, too, that the actual extent of transformation sharing and re-use is a function of the data mart architecture. In a standalone data mart, transformation sharing remains within the domain of the PowerCenter repository. Within a distributed architecture of multiple, linked data marts, the transformation object can be promoted up to the Global Repository, to be made available to developers across the enterprise.

## **3.4. PowerCenter Server Manager Module**

The Server Manager is a client tool used to schedule, execute, and monitor sessions and batches that perform the source to target data loads. The Server Manager allows developers to navigate through multiple folders and repositories, and monitor multiple Informatica Servers. The tool is a scheduler window for the different sessions and batches, which are typically composed of mappings created in designer. The Server Manager gives the data warehouse administrator control of the extracting and loading process. With the Server Manager the administrator can setup the production and/or test load window of the data warehouse. In many cases this window is overnight when server and database usage is light. The Server Manager will be used in the SFA project to setup the ETL window, schedule ad hoc jobs, and manage the development and test loads. Some of the features of the Server Manager:

- An advanced scheduler supports business scheduling and advanced error recovery, which adds greater detail to the error statistics maintained in the repository.

- New in PowerCenter 1.7: developers can create custom schedules to run sessions at multiple dates in a given month.

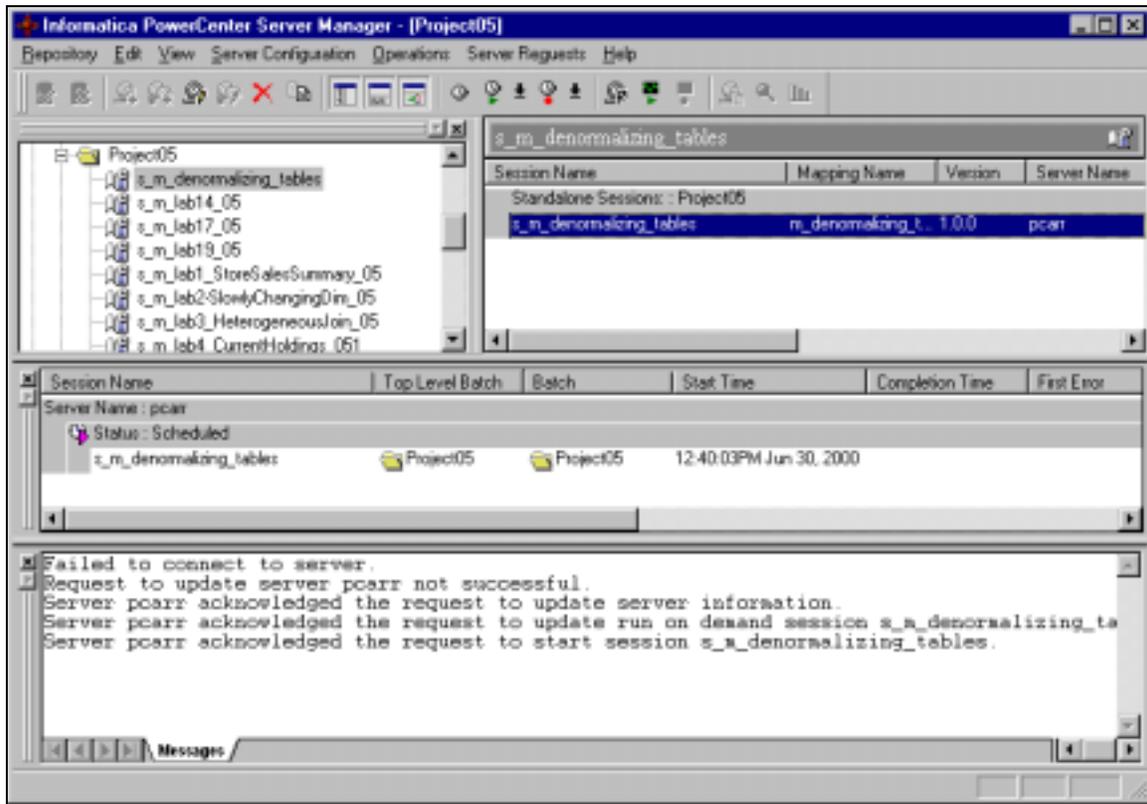


Figure 6 – Server Manager Module Window - Open

### 3.5. PowerCenter Repository Manager Module

The PowerCenter Repository is the metadata integration hub of PowerCenter 1.7. Developers gain access to the metadata stored in the repository through the Repository Manager and the Metadata Browser. A picture of the Repository Manager is below:

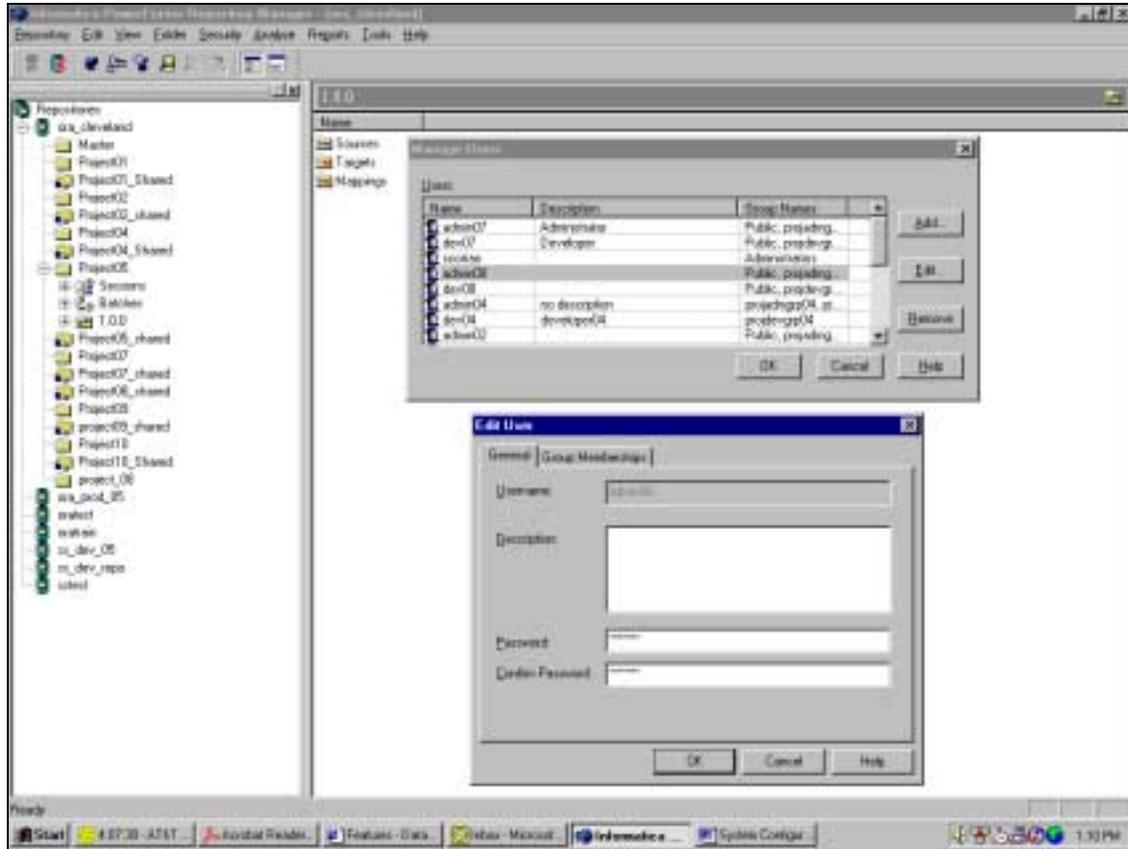


Figure 7 – Repository Manager Module - Open

The Repository Manager is used to create and maintain the PowerCenter repository and its metadata. Within the repository are three levels of detail:

### Folders

The highest-level logical groupings of data.

Examples are Customer, Vendor, Product, and Human Resources.

Permit sharing through Shared Folders, which may contain custom groupings of shareable transformations.

### Mappings

The business rules that source data must follow as it populates the data mart.

One or more mappings may be included within a Folder.

Examples are fact table loads, dimension table loads, and aggregate table calculations

### Objects

The lowest level of detail with which developers interact

Examples are source tables, data mart tables and transformations

The Repository Manager supports three types of management functions:

- *Top-level management* - Includes creation and maintenance of the repository (by connecting to, creating and deleting the repository, and, for Enterprise Data Mart developers, for registering with the PowerCenter Global Repository);
- *Folder management* - Includes creating, deleting, unlocking and, for Enterprise Data Mart developers, promoting Shared Folders to the Global Repository (for enterprise-wide sharing);
- *Security management* - Includes setting developer and group-level permissions, and maintaining folder and repository-level protections.

The Repository Manager also includes the Metadata Browser, which lets developers view metadata by browsing a Windows-like graphical tree structure, filtering and searching objects by various criteria. (In PowerCenter-built distributed data marts, the browser is also able to view the Shared-Folder contents of the Global Repository and other registered local data mart repositories.)

With the Metadata Browser, developers can search lists of tables and operational databases by a variety of criteria, and they can employ pop-up windows to analyze dependencies among operational source tables, data mart target tables, and transformation mappings.

The Metadata Browser offers standard data mapping and reference reports, and it lets developers create custom reports with third-party tools.

## 4 RDBMS Overview

### 4.1 Design Issues

Delivering a successful DWA requires a careful balance of three important factors: user requirements, query response time, and maintenance overhead. Clearly there are tradeoffs among these; for example, one could create aggregate tables to improve query response time, but at the expense of additional database maintenance. There will be a myriad of design decisions throughout the project. Each of these decisions should be made with an understanding of how these three factors are affected. Members of the project team should have a clear idea of the relative priority of each of the three factors.

The DBA staff is responsible for a number of critical tasks in the warehousing project. Skilled DBAs have practical experience in managing production databases. They have experience in RDBMS-specific configuration, backup and recovery strategies, performance tuning, system monitoring, data loading and storage strategies. In a business intelligence application, DBAs should also have an understanding of multidimensional (star schema) design concepts, query access patterns, indexing strategies, partitioning techniques, and pre-aggregation strategies.

### 4.2 Sizing and Configuration

The sizing and configuration of a data warehouse (and Oracle) are dependent on the applications that are to be supported. The number of users, amount of data, and query profile are all factors that must be considered in the sizing and configuration strategies. The designers of the individual applications must define each of these factors so that the data warehouse administrators can configure the RDBMS appropriately.

### 4.3 Data Model

Data warehouse applications are a fundamentally different breed of database application. Unlike transaction processing applications that take advantage of well-understood business processes, data warehouse systems add value by allowing innovative ad hoc analysis of information. It is essential that the project team build a business intelligence application on top of a database designed to accommodate this valuable type of analysis. Given the ad hoc nature of data access a business intelligence database must be designed to:

- Be intuitive for end users
- Provide good performance

## **4.4. Back-up and Recovery**

Like any production database, the data warehouse database and all other repositories required for the DWA should have a back-up and recovery plan. All repositories hold critical information and must be backed-up on a frequent basis. For the data warehouse however, a traditional back-up may not be the best method for recovery. It may be easier to extract, transform, and load data from the source systems than maintain a back-up and recovery system. Application developers must evaluate the various alternatives for back-up and recovery to manage the safety of the data and the costs of administration.

## **4.5. Data Load**

Resources for data extraction and transformation are characteristically underestimated in data warehousing projects. In addition, the complexity of the transformation effort increases exponentially with the number of data sources that must be integrated. Successful data warehouse projects are aware of the challenges of data transformation and are prepared to allocate resources in the event of unexpected challenges. To be completely successful, the transformation routines must be production ready and accomplish all necessary data extract and transformation tasks. The batch processes must complete all processing in the allotted batch window and require no manual intervention once in production. It is up to the application developers to manage the data load process.

## 5 MicroStrategy Overview

### 5.1 Introduction

MicroStrategy provides a suite of applications to enable Business Intelligence applications. Each application provides unique features and capabilities, yet all are integrated through a common metadata repository. Through the MicroStrategy Suite, users can receive analytical reports from the data warehouse through a client / server environment, over the web, e-mail, or wireless device. Currently MicroStrategy Intelligence Server, MicroStrategy Web, MicroStrategy Desktop and the MicroStrategy Metadata repository are installed and configured. Future application will utilize MicroStrategy Broadcaster and MicroStrategy InfoCenter.

### 5.2 MicroStrategy Intelligence Server

MicroStrategy Intelligence Server is the focal point of the MicroStrategy Platform. It is the middle-tier between the user applications and the data warehouse and provides report cache management, advanced analytical functions, job prioritization, and thread management.

### 5.3 MicroStrategy Metadata repository

All information for a MicroStrategy Project is stored in the metadata repository. The metadata repository contains detailed information on the data warehouse tables, attributes, facts, and relationships. All reporting objects including templates, filters, and reports for all users are also stored in the metadata repository. This central metadata repository architecture allows for reports to be created once and deployed through any of the MicroStrategy applications.

### 5.4 MicroStrategy Desktop

MicroStrategy Desktop is a client / server application which is the interface which allows access to MicroStrategy Architect, MicroStrategy Agent, and MicroStrategy Administrator.

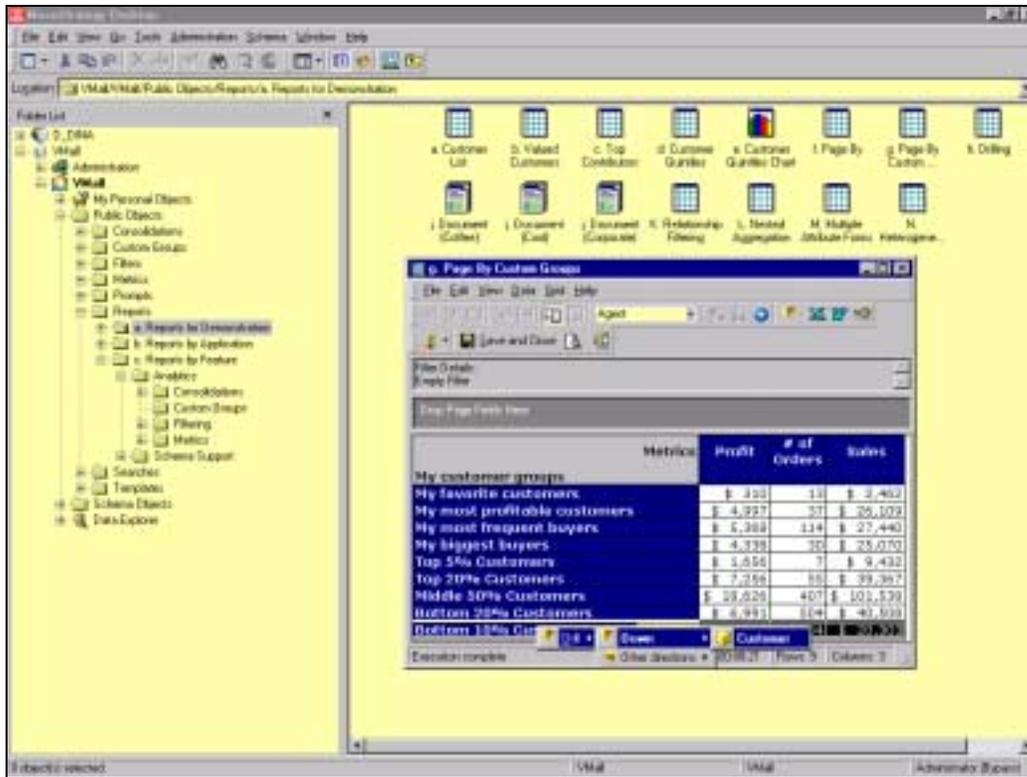


Figure 8 – MicroStrategy Desktop - Open

### 5.4.1. MicroStrategy Architect

MicroStrategy Architect is an OLAP application development tool. It is used to create a MicroStrategy project by identifying the data warehouse location, attributes, attribute hierarchies, facts, and relationships. Developers are also able to map business rules to the data in the data warehouse.

### 5.4.2. MicroStrategy Agent

MicroStrategy Agent is the most functionally rich OLAP tool. It provides OLAP functionality including report drilling and pivoting along with complex statistical, financial, and OLAP functions. Developers and power users can use this tool to create reports, define business metrics, and perform data analysis.

### 5.4.3. MicroStrategy Administrator

MicroStrategy Administrator contains two main features. First, it is used to manage MicroStrategy projects and users. Reporting objects can be managed between users and between a project's development, test, and production environments. User security and permissions can also be assigned to user groups, individual users, and individual reporting objects. The second main feature of MicroStrategy Administrator is Warehouse Monitor. Warehouse Monitor provides the ability to monitor and track project usage, report usage, report cache usage, warehouse threads and many other aspects of a project. The tool assists

administrator and developers in improving the overall performance of OLAP applications and to identify reports and projects that are successful or need improvement.

## **5.5. MicroStrategy Web**

MicroStrategy Web provides OLAP over the Internet through a web browser. It contains a majority of the features found in MicroStrategy Agent including report drilling, pivoting, and report creation. The user interface provided by MicroStrategy Web is created in pure HTML and is completely customizable.

## **5.6. MicroStrategy Broadcaster**

MicroStrategy Broadcaster pushes analytical data warehouse content beyond a client / server or web environment. Services (a collection of one or more reports) created in MicroStrategy Broadcaster can deliver analytical reports with information from the data warehouse to a variety of outputs including e-mail, wireless devices, and fax machines. Services can be sent out on a scheduled basis (the first of each month) or an alert basis (if expenses exceed \$500,000). Information delivered to each user is personalized so that only pertinent information is received.

## **5.7. MicroStrategy InfoCenter**

MicroStrategy InfoCenter works with MicroStrategy Broadcaster to allow users to subscribe to broadcast services through an Internet portal. MicroStrategy InfoCenter will develop a custom web portal that will allow users to subscribe to a list of available services, identify how they wish to receive the services, and apply criteria for personalization. Through MicroStrategy InfoCenter, users can also retrieve services in their personal My Documents portal. Once users elect to receive services through their My Documents portal, they can go to the web site to view the services. Each service available in My Documents is secure so that it is only available to the user to who subscribed to the report.

## 6 Data Warehouse Services

### 6.1 Metadata Lookup

Each application that requires the ability to view data source and transformation information will be able to access this information through MicroStrategy products. MicroStrategy Web or MicroStrategy Agent can be used to access the metadata information in the Informatica repository. Each data warehouse application that is built will contain reports that query the metadata and will provide the information to the user.

### 6.2 ROLAP (Client / Server)

Through an internal client/server environment, users can analyze and build analytical reports on data in the data warehouse. Through MicroStrategy Desktop or a custom application using the MicroStrategy Software Development Kit (SDK), users can be given the functionality of an OLAP power user tool. Based on the particular user or customized application, functionality can be extended or limited to meet specific requirements.

In order to access the data warehouse in a client / server environment, MicroStrategy Agent or a custom application using MicroStrategy SDK must be used and installed on the client machine. The MicroStrategy SDK is a complete COM based API that will enable analysis, reporting, administration, security and management of an OLAP system.

### 6.3 ROLAP (Web)

Internal or external users can access OLAP functionality to analyze data in the data warehouse over an Intranet or Internet through a web browser. Through MicroStrategy Web or a custom web application using the MicroStrategy SDK, users can perform analytical reporting and analysis. Based on a particular user or custom application, functionality can be extended or limited to meet specific requirements.

Viador will establish the connection to MicroStrategy Web for access to OLAP services. Viador will call an active server page (ASP) page via uniform resource locator (URL) on the MicroStrategy Web machine with, at minimum, parameters for the project name and user login. Additional application specific parameters may be passed such as user group, report name, error page URL, and return page URL. It is assumed that users have been authenticated before the ASP page is called. The ASP page on the MicroStrategy Web machine will then determine the correct MicroStrategy project name, project URL and server instance needed to access the specific MicroStrategy Web application.

Developers will be provided the following:

- ASP page URL
- Input parameter name and requirements

## **6.4. DSS Administration**

Administration of the DWA is necessary in order to maintain and build applications. Administration functions of a decision support system) include managing and organizing users, groups, projects, and database connections; coordinate and prioritize user requests; allocate the resources necessary to complete user requests; create schedules and manage schedule requests; manage security; and monitor and analyze the daily activity of the system. MicroStrategy Administrator will be used through MicroStrategy Desktop to administer and manage DSS projects.

All administration functions for the DSS environment can be done through MicroStrategy Administrator or a customer application using MicroStrategy SDK. Either of these applications must be installed on a client machine.

## **6.5. Extracting**

The extract step is the first step of getting data into the data warehouse environment. Extracting means reading and understanding the source data, and copying the parts that are needed to the data staging area for further work.

## **6.6. Transforming**

Once the data is extracted into the data staging area, there are many possible transformation steps, including:

- Purgig selected fields from the legacy data that are not useful for the data warehouse
- Cleaning the data
- Combining data sources
- Building aggregates for boosting the performance of common queries

## **6.7. Loading and Indexing**

At the end of the transformation process, the data is in the form of load record images. Loading in the data warehouse environment usually takes the form of replicating the dimension tables and fact tables and presenting these tables to the bulk loading facilities of each recipient data mart.

## **6.8. Quality Assurance Checking**

Running a comprehensive exception report over the entire set of newly loaded data can check quality assurance. The exception report can be built using the data mart's end user report facility.

## **6.9. Release/Publishing**

When each data mart has been freshly loaded and quality assured, the user community must be notified that the new data is ready. Publishing also communicates the nature of any changes that have occurred in the underlying dimensions and new assumptions that have been introduced into the measured calculated facts.

## **6.10. Updating**

Overtime, the data warehouse and accompanying data marts may require updates. Changes in labels, change in hierarchies, change in status, and changes in corporate ownership often trigger necessary changes in the original data stored in data marts that comprise the data warehouse, but in general these are managed load updates, not transactional updates.

## **6.11. Querying**

Querying include report writing, complex decision support applications, requests from models, and full-fledged data mining. Querying never takes place in the data staging area. By definition, querying takes place on a data warehouse presentation server.

## **6.12. Data feedback**

There are two important places where data flows “uphill” in the opposite direction from the traditional flow. First, we may upload a cleaned dimension description from the data staging area to a legacy system. This is desirable when the legacy system recognizes the value of the improved data. Second, we may upload the results of a complex query or a model run back into a data mart. This would be a natural way to capture the value of a complex query that takes the form of many rows and columns that the user wants to save.

## **6.13. Auditing**

At times it is critically important to know where the data came from and what calculations were performed.

## **6.14. Securing**

Data warehouse security must be managed centrally, from a single console. Users must be able to access all the constituent data marts of the data warehouse with a single sign-on.

## **6.15. Backing Up and Recovering**

Since data warehouse data is a flow of data from the legacy systems on through to the data marts and eventually onto the users’ desktops, a real question arises about where to take the

necessary snapshots of the data for archival purposes and disaster recovery. Additionally, it is necessary to back up all of the metadata.

## 7 Data Warehouse Naming Conventions

### 7.1 Introduction Informatica PowerCenter Standards

Informatica PowerCenter provides graphical user interface (GUI) objects that are used to build mappings. It is recommended that developers adhere to the established naming standard when creating objects within a mapping. This allows for consistent conventions across the enterprise, but also enables a developer or user to identify a specific type of transformation simply by viewing that object's name in the metadata. The available transformations and the standard are shown in the following table.

Table 3 – Informatica PowerCenter Transformation Standards

Transformation Type	Recommended prefix	Example
Aggregator	AGG	AGG_SALES_PER_REGION_ITEM
Expression	EXP	EXP_CALCULATE_REVENUE
External Procedure	EXT	EXT_COM_PROGRAM
Filter	FIL	FIL_HIGH_SALARIES
Joiner	JNR	JNR_JOINER
Normalizer	NORM	NORM_FIL_ONE
Rank	RNK	RNK_EMPLOYEES
Lookup	LKP	LKP_ZIP_CODE
Sequence Generator	SEQ	SEQTRANS
Source Qualifier	SQ	SQ_CUSTOMERS
Stored Procedure	SP	SP_GET_LAST_NAMES
Update Strategy	UPD	UPD_SOC_SEC_NUM

### 7.2 MicroStrategy Naming Conventions

Generally, the naming conventions used in the MicroStrategy project are application specific. However there are several naming techniques that are recommended for ODBC data source names (DSN). The recommended DSN are as follows:

- ODBC DSN connection to the data warehouse – *database name\_WH*
- ODBC DSN connection to the metadata repository – *database name\_MD*

### 7.3 Informatica Directory Structures

The tables below list and describe the contents of the directories that were created to install the Informatica server and client. For the installation of the development environment at the VDC, the default directories were used. Note that the server was installed on Sun Solaris, while the client was installed on Windows NT 4.0.

#### 7.3.1 PowerCenter Server

Table 4 – Directory Structure – PowerCenter Server: Unix

Directory	Contents
/home/[user]/Informatica/PowerCenter	Contains exe. files for configuring the server, plus error log files
/home/[user]/Informatica/PowerCenter/BadFiles	Contains reject files
/home/[user]/Informatica/PowerCenter/Cache	Default directory for lookup cache, index and data caches and data files
/home/[user]/Informatica/PowerCenter/ExtProc	Default directory for external procedures
/home/[user]/Informatica/PowerCenter/SessLogs	Contains session log files
/home/[user]/Informatica/PowerCenter/SrcFiles	Default directory for source files
/home/[user]/Informatica/PowerCenter/TgtFiles	Default directory for target files

#### 7.3.2 PowerCenter Client

The PowerCenter Client is available on Windows 95, Windows 98, and Windows NT. On all of these platforms, the directory structure is as follows:

C:/Program files/Informatica/PowerCenter Client/Workspace

### 7.4 MicroStrategy Directory Structures

The table below lists and describes the contents of the root Windows directories which will be created when MicroStrategy products are installed. Following this table are the specific directories structures for each of the MicroStrategy products. For more information on the directories and the files in the directories, please see the MicroStrategy product manuals.

Table 5 – Directory Structure - MicroStrategy: Windows

Directory	Contents	Size
\Program Files\MicroStrategy\	Root MicroStrategy 7.0 directory	See below
\Program Files\Common Files\MicroStrategy\	Common MicroStrategy 7.0 Files	71 MB

### 7.4.1. MicroStrategy Intelligence Server 7.0

Table 6 –Directory Structure – MicroStrategy Intelligence Server 7.0

Directory	Contents	Size
\Intelligence Server\	Root MicroStrategy Intelligence Server directory	5.2 MB
\Intelligence Server\Inbox\	Report files	0 MB
\Intelligence Server\Log\	Log files	0 MB

### 7.4.2. MicroStrategy Web 7.0

Table 7 – Directory Structure – MicroStrategy Web 7.0

Directory	Contents	Size
\Web\	Root MicroStrategy Web directory and Interface ASP files	6.8 MB
\Web\Admin \	MicroStrategy Web Administrator Directory	53 KB
\Web\Corelib\	MicroStrategy Web Core Library Files	438 KB
\Web\Customlib\	MicroStrategy Web Custom Library Files	1.1 MB
\Web\Help\	MicroStrategy Web Help files	2.2 MB
\Web\Images\	MicroStrategy Web Image Files	1 MB
\Web\Styles\	MicroStrategy Web CSS Files	100 KB
\Web\Internationalization\	MicroStrategy Web International Files	321 KB

### 7.4.3. MicroStrategy Desktop 7.0

Table 8 – Directory Structure – MicroStrategy Desktop 7.0

Directory	Contents	Size
\Program Files\MicroStrategy\Desktop	MicroStrategy Desktop root directory	76 MB
\Program Files\MicroStrategy\Desktop\Images	Images used in the Desktop Homepage	334 KB
\Program Files\MicroStrategy\Desktop\Log	Log directory, for Diagnostics output	0
\Program Files\MicroStrategy\Desktop\XSLs	Autostyles for grids2	721 KB

## 8 Data Warehouse Application Design

### **8.1. Introduction**

There are several application design principles and best practices that can be used to assist application developers.

### **8.2. Report Interfaces for Users**

There are several methods in which users can access reports through the DWA. Depending on user needs one or more of the methods would be used. The following is a list and descriptions for the various methods:

#### **8.2.1. ROLAP (Web)**

MicroStrategy Web is typically used to deploy ad-hoc reporting and data analysis to a large number of users through an Internet or Intranet environment.

#### **8.2.2. ROLAP (Client / Server)**

MicroStrategy Agent is generally reserved for business analysts and power users who would be performing sophisticated high-end report development and analysis.

## 9 Data Warehouse Programming Choices

### 9.1 Introduction

Custom development can be done within the DWA. In both Informatica and MicroStrategy application, APIs are available for customization or application development. The following is an overview of several optional programming choices that are available. Additional information on these is available in the Appendix.

### 9.2 Informatica

#### 9.2.1 External Procedure Transformations

External Procedure transformations operate in conjunction with procedures created outside of the Designer interface to extend PowerCenter functionality. Although the standard transformations provided offer a wide range of transformation options, there are occasions where extended functionality may be desired. For example, the range of standard transformations (Expression, Stored Procedure, Filter, and so forth) may not provide the exact functionality desired. An experienced programmer may want to develop complex functions within a dynamic link library (DLL) or Universal Interactive Executive (UNIX) shared library, instead of creating the necessary transformations in a mapping.

#### 9.2.2 Informatica PowerCenter.e

In addition to the optional programming features provided with PowerCenter, PowerCenter.e has a number of similar features as well. PowerCenter.e is an add-on component that enables the ETL process to integrate web-based data from multiple channels into the data warehouse. The tool works in conjunction with PowerCenter and existing e-business data management products, to perform lookups, data transformations, and data analysis of common web-based file formats. PowerCenter.e includes the following optional tools that allow for sourcing data from the following:

- Extensible Markup Language (XML)
- Message queue from IBM's Message Queuing (MQ) series
- Standard and extended web log formats
- Demographic lookups from Axiom Data Network (ADM)

PowerCenter.e also includes reusable transformations for:

- Sorting file data
- Calling Perl functions from PowerCenter mappings

### **9.3. MicroStrategy Programming Choices**

MicroStrategy production provide APIs in the following areas:

- MicroStrategy Intelligence Server
- MicroStrategy Web
- MicroStrategy Desktop 7.0

#### **9.3.1. MicroStrategy Intelligence Server, MicroStrategy Web, and MicroStrategy Desktop 7.0**

All functionality available in the MicroStrategy 7.0 suite is exposed in the MicroStrategy SDK 7.0. This set of APIs is COM based and can be utilized by client/server and Internet application. For more information on the MicroStrategy SDK 7.0, please see the MicroStrategy 7.0 Developer guides.

## 10 Data Warehouse Configuration

### **10.1. Introduction**

This section describes the important considerations when designing and configuring the DWA components listed below.

- PowerCenter Client Workstation
- PowerCenter Server
- PowerCenter Repository
- MicroStrategy Desktop (MicroStrategy Agent, MicroStrategy Architect, MicroStrategy Administrator) 7.0
- MicroStrategy Intelligence Server 7.0
- MicroStrategy Web 7.0
- Metadata Repository
- MicroStrategy Server Hardware Sizing

## **10.2. PowerCenter Client Workstation**

The tools installed on the client workstation include both Informatica and MicroStrategy application.

### **10.2.1. Installation Prerequisites**

The following are minimum requirements to install the Informatica Client:

- 32 MB RAM
- Operating System Windows 95/98 or NT 4.0 with service pack 3 or 4
- 40 MB Disk space
- The windows temp variable must point to a valid directory
- The login must have administrator rights on the client
- 32 bit database client utilities installed for repository, all sources and targets
- Client can access Oracle via Net 8 and sql plus; Informix via Embedded SQL (ESQL)/C and Interactive SQL (ISQL), DB2 via DB2 connect and Dbaccess, etc.
- environment or regional setting (code page) compatible with server and repository
- TCP/IP configured to communicate with the PowerCenter Server host

### **10.2.2. Installation and Configuration Guidelines**

Generally, the default options should be used throughout the installation process. Attention should be given to the Setup options and the Selected Components.

#### **Setup options**

During setup, the user then can choose a different location to install the application and files or accept the default c:/program files/Informatica/PowerCenter Client directory.

#### **Selected Components**

During setup, a user will be given the option to select any of or all of the following components:

- Informatica server
- Client
- ODBC

## **10.3. PowerCenter Server**

### **10.3.1. Installation Prerequisites**

The following questions needed to be addressed prior to installing the server.

- Is operating system Sun Solaris 2.6 or 2.7?
- Is sufficient memory available? (2 gig RAM or more recommended)
- How much disk space is available? (5-10 gig recommended for FTP files)
- Are CD's available for Oracle, DB2 Connect, DB2, Informix, and any databases to be used (including database drivers, query tools, and documentation)?
- Is environment or regional setting (code page) compatible with server and repository?
- Can each database be queried using the query tools?
- Are CDs the same version of drivers installed on client and DBMS host server?
- Are products supported on availability matrix for server compatibility / support?

## 10.4 PowerCenter Repository

### 10.4.1 Prerequisites

Before creating a repository, verify the system has the following software installed and sufficient disk space and memory:

- Windows NT 4.0 or UNIX operating system
- 60MB disk space (60-120 MB is recommended)
- ODBC drivers or native drivers

When creating a repository, the developer must have the following information available:

- Data source to connect to the database.
- Database username and password. This login must have the appropriate database permissions to create the repository. In the new repository, this login becomes a default user with full privileges in the repository. The username might be in other languages, but the password must be in US-ASCII only.
- Code page. Contains the character set of the data in the repository. Once specified, the code page cannot be changed.

The developer must have one of the following repository privileges:

- Administer Repository
- Super User

Before the developer can create a repository, the database to contain the repository must first be created and configured. You can locate the repository on the source or target database systems. However, to protect your repository, consider keeping the repository separate from overloaded machines.

A database can have only one repository with the same database username. If the developer creates a new repository in a database with an existing repository, the existing repository is overwritten and loses all metadata within it. The Repository Manager detects any existing repositories and offers the developer the opportunity to cancel the new repository or delete the existing one.

When using a PowerCenter Repository Manager to create a new repository, the Repository Manager offers the option of creating a local or global repository. After creating a local repository, the developer can promote it to a global repository. However, once a global repository is created, it cannot be changed to a local repository.

This section identifies minimum requirements for client workstation expecting to access the Enterprise Data Warehouse (EDW) from within this technical architecture.

## **10.5. MicroStrategy Desktop (MicroStrategy Agent, MicroStrategy Architect, MicroStrategy Administrator) 7.0**

### **10.5.1. Prerequisites**

This section discusses the installation prerequisites and installation and configuration guidelines for MicroStrategy Desktop 7.0 (MicroStrategy Agent, MicroStrategy Architect, and MicroStrategy Administrator). These client tools are needed for the development, test, production, and operations environments.

#### **Client Server Applications**

The minimum workstation hardware and software for client/server applications must include:

- Pentium 266 MHz CPU
- 64 MB RAM
- TCP/IP network protocol
- Microsoft (MS) Windows 95 or greater
- 256 color palette or more
- 800x600 resolution or greater

#### **Web Applications**

The minimum workstation requirements for web applications are an HTML 2.0 compatible web browser and the hardware required for the web browser application. The recommended hardware for clients accessing web applications is as followed:

- Pentium 133 MHz CPU
- 32 MB Ram
- TCP/IP Protocol
- MS Windows 95 or grater
- 256 color palette or more
- 800x600 resolutions or greater

### **10.5.2. Installation Prerequisites**

The following are minimum requirements to install MicroStrategy Desktop:

- Windows 95, 98, NT 4.0 SP4
- TCP/IP network protocol

- 256 colors
- DCOM installed (installed automatically if it is not already installed)
- MS Internet Explorer (IE) 4.01 Service Pack 1

If the installed version of Microsoft Internet Explorer is lower than that required, than the Installation Wizard provides the option to install Microsoft Internet Explorer 5.0.

### **10.5.3. Installation and Configuration Guidelines**

Generally, the default options should be used throughout the installation process. Setup Options and Selected Components are two steps within the installation process that allow the user to specify how he or she would like to install the product.

#### **Setup options**

The two Setup options that are available are Typical and Advanced. In a Typical Setup, all selected MicroStrategy Products are placed on the same drive and the system assigns the common file location. In Advanced Setup, the developer can select a different drive for each selected MicroStrategy Product and select a location for common files.

*Recommendation:* When installing two or more MicroStrategy Products on a given machine and multiple drives are available, then the Advance Setup should be used in order to place each product on its own drive. Otherwise, Typical Setup should be used.

#### **Selected Components**

When installing MicroStrategy Desktop, you may choose which components are installed. The developer may choose one or more of the following components:

- MicroStrategy Agent
- MicroStrategy Architect
- MicroStrategy Administrator

A developer can verify installation setup information through the installation log file (install.log) located by default in C:\Program Files\Common Files\MicroStrategy. The installation log file can be particularly helpful if errors are encountered during the installation process.

## 10.6. MicroStrategy Intelligence Server 7.0

This section discusses the installation prerequisites and installation and configuration guidelines for MicroStrategy Intelligence Server 7.0.

### 10.6.1. Installation Prerequisites

The following are minimum requirements to install MicroStrategy Intelligence Server 7.0:

- Windows NT 4.0 SP4
- TCP/IP network protocol
- 3 MB of memory for registry
- 256 colors
- Microsoft Internet Explorer 5.0
- Microsoft Data Access Components (MDAC) 2.1 SP2 (Installed automatically)
- Certified ODBC drivers for data warehouse and metadata repository (specific drivers depend on the RDBMS used)

### 10.6.2. Installation and Configuration Guidelines

In order to install and configure MicroStrategy Intelligence Server 7.0, the user login must have Windows NT administrator privileges for the domain or target machine. The domain must include the target databases (data warehouse and metadata repository).

During installation, all applications must be closed including the MS Office Shortcut Bar and NT Control Panel, and all services as feasible. This precaution is necessary to ensure proper registration of files. Generally, the default options should be used throughout the installation process. Setup Options and Service Account Name are two steps within the installation process that allow the user to specify how he or she would like to install the product.

#### Setup options

The two Setup options that are available are Typical and Advanced. In a Typical Setup, all selected MicroStrategy Products are placed on the same drive and the system assigns the common file location. In Advanced Setup, the developer can select a different drive for each selected MicroStrategy Product and select a location for common files.

*Recommendation:* When installing two or more MicroStrategy Products on a given machine and multiple drives are available, then the Advance Setup should be used in order to place each product on its own drive. Otherwise, Typical Setup should be used.

### **MicroStrategy Intelligence Server Service Account Name**

Because MicroStrategy Intelligence Server runs as a Service, a service login and password is required. The login provided for the service must be a Windows NT account with administrative privileges.

A developer can verify installation setup information through the installation log file (install.log) located by default in C:\Program Files\Common Files\MicroStrategy. The installation log file can be particularly helpful if errors are encountered during the installation process.

Please see the MicroStrategy Product installation manuals for more detailed information concerning software installation and configuration.

Businesses can implement clustering and fail-over support specific user requirements. Multiple MicroStrategy Intelligence Servers can be clustered to increase scalability & performance and implement fail-over support. These MicroStrategy Intelligence Servers share caches and perform object management across the cluster so that system integrity and efficiency is maintained.

## **10.7. MicroStrategy Web 7.0**

This section discusses the installation prerequisites and installation and configuration guidelines for MicroStrategy Web.

### **10.7.1. Installation Prerequisites**

The following are minimum requirements to install MicroStrategy Web:

- Windows NT 4.0 Service Pack 4
- TCP/IP network protocol
- 3 MB of memory for registry
- 256 colors
- Microsoft Internet Explorer 5.0
- Microsoft Internet Information Server 4.0

If the installed version of MS Internet Explorer is lower than that required, than the Installation Wizard provides the option to install Microsoft Internet Explorer 5.0.

### **10.7.2. Installation and Configuration Guidelines**

In order to install and configure MicroStrategy Web, the user login must have Windows NT administrator privileges for the domain or target machine. The domain must include the target databases (data warehouse and metadata repository).

During installation, all applications must be closed including the MS Office Shortcut Bar and NT Control Panel, and all services as feasible. This precaution is necessary to ensure proper

registration of files. Generally, the default options should be used throughout the installation process. Attention should be given to the Setup options and the MicroStrategy Web setting.

### **Setup options**

The two Setup options that are available are Typical and Advanced. In a Typical Setup, all selected MicroStrategy Products are placed on the same drive and the system assigns the common file location. In Advanced Setup, the developer can select a different drive for each selected MicroStrategy Product and select a location for common files.

Recommendation: When installing two or more MicroStrategy Products on a given machine and multiple drives are available, then the Advance Setup should be used in order to place each product on its own drive. Otherwise, Typical Setup should be used.

### **MicroStrategy Web setting**

A name must be given to the Microsoft IIS virtual directory that is to be created. This name should be meaningful to developers who may access the server machine.

A developer can verify installation setup information through the installation log file (install.log) located by default in C:\Program Files\Common Files\MicroStrategy. The installation log file can be particularly helpful if errors are encountered during the installation process.

Businesses can implement clustering and fail-over support at two levels in the architecture. Multiple MicroStrategy Web Servers can be clustered using a hardware solution (such as one from Cisco) or a software solution (such as Windows LBS or Resonate). The MicroStrategy Web Server in turn has a load balancing and fail-over component.

## **10.8 Metadata Repository**

The MicroStrategy metadata repository can be created in MS SQL Server, Oracle, or IBM DB2. Configuration of the metadata repository will depend on the specific RDBMS that is used. The MicroStrategy Intelligence Server Configuration Wizard creates the physical metadata tables when a project is created.

Because the MicroStrategy Metadata contains all the information for a project, it is necessary to backup the database on a regular basis.

## 10.9. MicroStrategy Server Hardware Sizing

This section will discuss each product and how the product should be sized given a set of requirements. Hardware requirements for MicroStrategy products depend on many factors. For a given environment, these factors will determine requirements for:

- CPU
- Memory
- Hard disk space

CPU calculations are very site / application specific. Although it is possible to create reasonable metrics for an established application, it's nearly impossible to accurately predict the CPU requirements for an application that has not yet been designed. In the absence of this information, we must resort to similar experiences with other customers with similar user and data capacities to produce a rough estimate. The memory and disk requirements for production components can be reasonably estimated given the site / application requirements and certain assumptions regarding the system configuration. These estimates will only be as accurate as the assumptions and capacity estimates themselves and must be refined as the architecture and application designs mature.

### 10.9.1. MicroStrategy Web and Intelligence Server 7.0 Sizing

The minimum hardware requirements for MicroStrategy Web 7.0 and MicroStrategy Intelligence Server 7.0 are a Pentium II, 450 MHz processor, 128 MB RAM, and 200 MB hard disk storage.

System sizing for MicroStrategy Web and Intelligence Server 7.0 depend on each of the following factors, number of users, report complexity, ad-hoc versus cache report requests and other considerations.

#### Number of users

The number of users in a system can be measured in several ways.

- **Total users** are the number of registered users in the system. For example, if a corporate web site is available to be viewed by 950 individuals, the site has 950 total users.
- **Active users** are those users who are logged into the system. For example, if a corporate web site is available to be viewed to 950 individuals and 30 of them are logged onto the site, there are 30 active users.
- **Concurrent users** are those users who have jobs (report requests) being processed by a server (MicroStrategy Web or MicroStrategy Intelligence Server) at the same time. For example, a corporate web site is available to be viewed by 950 different individuals, and 30 people are logged in. Of those 30 active users, 10 have jobs being processed by the server. These 10 users are considered concurrent users.

The number of concurrent users is the most important one to consider. The system must be able to support the maximum number of concurrent users that are expected at any given time.

Through Best-practices, the following table was developed that illustrates the ratio of Concurrent users to Total users in three application scenarios. The application scenarios are defined as followed:

- Typical enterprise system – Intranet application for reporting and analysis of internal data
- Typical e-Business system – Internet application that is available to customers (students, schools, or financial partners).

Table 9 – Application Scenario Concurrent Users to Total Users Ratios

Scenario	% of Concurrent Users to Total Users
Typical enterprise system	3.50%
Peak for a typical e-Business system	1.00%
Average for a typical e-Business system	0.35%

### Report Complexity

In the context of MicroStrategy product sizing, report complexity is measured in terms of the number of data cells for the report and the number or SQL passed needed to create the report. The more complex a report, the more stress on the system. Developers should have an idea of the level of report complexity that is expected from their applications and/or the expected ratio of complex reports to simple reports. As the number of complex reports increases, a developer must either implement a caching strategy or increase the systems processing power.

### Ad-hoc versus Cache Reports

MicroStrategy Web 7.0 caches reports, objects and elements by default. Caching allows users to experience better performance time while minimizing the load on the MicroStrategy Intelligence Server 7.0 and database server. Because simple reports do not require a significant amount of processing time or power, the largest benefits of caching will come from the caching of complex reports. The more complex reports that are cached, the less processing power will be required.

### Other Considerations

The sizing methodologies for MicroStrategy Web 7.0 and Intelligence Server 7.0 are based on the assumption that the majority of the users will access the system through MicroStrategy Web with only a small percentage of the users accessing the system through MicroStrategy Desktop.

Statistical logging is very useful when analyzing the system. However it does create additional load on the system and will reduce the overall system response times. It is therefore recommended that logging should only be turned on periodically.

Typically a system will have a 1:1 ratio of MicroStrategy Intelligence Servers 7.0 to MicroStrategy Web 7.0 Servers. However, a developer may find better performance by adding additional servers to either side depending on the particular requirements of the application.

The following tables will help to determine the best configuration for a system. There is no exact formula for determining hardware specifications because one is unable to predict actual system load until applications are developed deployed and monitored. These suggestions are intended to be basic guidelines to be used when the developer initially configures the system. Periodic evaluations are recommended to a system in order to update the configuration based on actual system performance and usage patterns.

The first table lists some possible hardware configurations for your system. All configurations have MicroStrategy Intelligence Server 7.0 and MicroStrategy Web 7.0 on different machines except for the \*Quad configuration, which has them on the same machine. For each configuration, processor speed, number of Processors and RAM is given. The number of MicroStrategy Intelligence Server and MicroStrategy Web Server machines are also given.

The second table shows which of the above configurations to use based on the number of Concurrent users and the desired system response time.

Table 10 – Potential Hardware Configurations

<b>Configuration</b>	<b>Processor Speed (MHz)</b>	<b>RAM (MB)</b>	<b>Number of Processors</b>	<b>MicroStrategy Intelligence Servers</b>	<b>MicroStrategy Web Servers</b>	<b>Total Machines</b>
Single	450	128	1	1	1	2
Dual	500	512	2	1	1	2
*Quad	500	1024	4	1	1	1
Quad	500	1024	4	1	1	2
Cluster	500	1024	4	2	1	3

Table 11 – Recommended Configuration Based on Number of Concurrent Users And Desired Response Time

Concurrent Users	Configuration		
	Minimum (response time less than 18 seconds)	Recommended (Response time of about 12 seconds)	Optimal (response time less than 10 seconds)
0-50	Single	Single	Single
50-75	Single	Dual	Dual
75-100	Dual	Dual	*Quad
100-150	Quad	Quad	Quad
150-200	Quad	Quad	Cluster
200-250	Quad	Cluster	

By combining the information from the two tables, developers are able to estimate the hardware that is required to support a given user population.

### 10.9.2. Metadata Repository Server Sizing

It is recommended that in a production environment the metadata repository be stored on its own machine with no other databases. This is due to the critical nature of the data and to provide maximum performance. Sizing of the metadata repository database depends on the number of reporting objects created for all of the projects in the metadata database. It is expected that the database will not exceed 5 GB in size. However, it is recommended that the size of the database be monitored periodically.

The server requirements depend greatly on the RDBMS used. Although the MicroStrategy metadata is small (typically less than 1 GB), it is recommended that an Oracle DBA be consulted to determine an optimal hardware configuration.

## 11 Data Warehouse Security Architecture

### 11.1 Introduction

Each individual component within the DWA handles security. However, security can be setup so that all components work together.

### 11.2 Informatica Security

#### 11.2.1 Introduction

The Informatica Client, Server, and repository offer several layers of security that can be used to customize the repository and data warehouse. The following features are available:

- **User groups.** Repository groups for usernames. Users can be assigned to multiple groups. Privileges are assigned to groups; every user in the group receives privileges for that group. These groups can also be used to handle Owner's Group folder permissions.
- **Repository users.** Username is used to access the repository. You can assign privileges to individual usernames, though each username must have a unique repository username to use folder and object locking properly. Each user must be assigned to at least one group.
- **Repository privileges.** The ability to perform actions within the repository and to start and stop the Informatica Server. You assign repository privileges to users and groups. Even though repository privileges have been granted to perform certain tasks in the repository, a user may also require permission to perform tasks in a given folder.
- **Folder permissions.** The ability to perform tasks within an individual folder. Permissions can be granted on three levels: to the folder owner, a group to which the folder belongs, and the rest of the repository users.
- **Locking.** The repository locks lock repository objects and folders by the user. The repository creates five kinds of locks depending on the task performed: read, write, execute, fetch, and save. Locks can be broken, but to avoid repository inconsistencies it should be determined if the owner of the lock is actually using the object.

Note that in addition to the Informatica-specific security features, developers will need to have appropriate RDBMS privileges (see section 14.3) for all procedures that require a call to the RDBMS (e.g. select, insert, update, etc.)

#### 11.2.2 Securing Development Environments

In a development environment, the developer should protect repository metadata. With multiple users editing and testing repository objects, developers are likely to encounter locked objects and folders. They may be locked due to a prior system problem, or because

another user is working with the object or folder. To prevent repository inconsistencies, determine who owns the lock and make sure that user is not using the object before overriding the repository lock feature.

Rather than tightening security at a later date, we recommend developers establish appropriate security measures while in development.

### **11.2.3. Securing Production Environments**

In a production environment, it is critical to implement strong security measures to protect your repository and data warehouse. For this reason, limit the number of users accessing the repository. Grant each user only the privileges and permissions necessary to perform their assigned tasks.

Avoid editing metadata in a production repository. Investigate all repository locks carefully before overriding them. You can unlock individual objects, folder versions, and folders in the Repository Manager. Unlocking objects, versions, or folders inappropriately can cause repository and data warehouse inconsistencies. Overriding a lock on an executing session or batch can cause inconsistencies in your data warehouse.

### **11.2.4. User Groups**

Custom user groups are created to manage users and repository privileges efficiently. After creating a new user group, a set of privileges are assigned to the group.

Each repository user must be assigned to at least one user group. When a user is assigned to a group, the user:

- Receives all group privileges
- Inherits any changes to group privileges
- Loses and gains privileges if the user group membership is changed

You can also assign users to multiple groups. This grants the user the privileges of each group.

#### **Default Groups**

When the developer creates a repository, the Repository Manager creates two repository user groups. These two groups exist so the developer can immediately create users and begin developing repository objects. However, the developer can also create custom groups and assign specific privileges and permissions to those groups.

There are two default repository user groups:

- Administrators
- Public

You cannot delete these groups from the repository or change their configured privileges. The Administrators group has full privileges within the repository. The Public group has a subset of default repository privileges.

The Repository Manager automatically creates two default users in the Administrators group:

- Administrator
- The database username used to create the repository

You cannot delete these users from the repository or remove them from the Administrators group. The Repository Manager does not create any default users for the Public group.

### **User Groups and Folder Permissions**

When a folder is created or edited, Owner's Group permissions for the folder can be defined. You can only grant Owner's Group permissions to one of the groups to which the folder owner belongs. If the owner belongs to more than one group, one of those listed groups must be selected to receive Owner's Group permissions.

If the owner of the folder belongs to both the Developer group and the Production group, Owner's Group permissions must be granted to one of the two groups. If the Production group is selected, the Production group receives read and write permissions on the folder.

### **Creating a User Group**

User groups are created in the Repository Manager. To create a user group, the developer must have one of the following privileges:

- Administrator Repository
- Super User

### **Editing a User Group**

At any time, the description of an existing user group can be edited. However the default groups (Public and Administrator) cannot be edited. To edit a user group, the developer must have one of the following privileges:

- Administrator Repository
- Super User

### **Deleting a User Group**

You can delete any user group in the repository except the default repository user groups, Public and Administrators.

To delete a user group, the developer must have one of the following privileges:

- Administrator Repository
- Super User

**Note:** If a group that contains users is deleted, the Repository Manager reassigns those users to the Public group.

## Repository Users

Each repository user needs a username and password to access the repository. When the repository is created, the repository creates two default users.

- Administrator (password: Administrator)
- Database user (the username and password of the user when the repository was created)

## Users and Locking

When a user works with a repository object, such as a reusable transformation, a mapping, or session, the repository locks the object. The repository uses different locks for different tasks performed on the object.

When the repository locks an object, it notes the username accessing the object. If another user attempts to perform the same task with the object, the repository issues a warning telling the second user that the object is locked.

If a user loses connection to the repository, the repository retains all of the user's locks. In these cases the user can reacquire the locks or unlock the locked objects as appropriate.

## Locking and Lost Connections

When a mapping is edited in the Designer, the repository creates a write lock on the mapping for the user. If your client machine crashes while editing the mapping, the repository keeps the mapping locked. When a developer reboots the client machine and attempt to open the same mapping at the same machine, the Designer issues the following warning:

The mapping [mapping\_name] is already locked by [your\_username]. Do you want to reacquire the lock?

Since the developer owns the lock and know why the object is already locked, then the lock can be safely reacquired.

## Locking and Duplicate Access

When a developer owns a write lock on a mapping, another user might attempt to edit the same mapping. In this case, if each user accesses the repository with separate usernames, the repository recognizes the mapping as locked by a different user. The Designer displays the correct lock owner in the message box, and the second user cannot reacquire the lock.

## Users and User Groups

When a username is created, the user must be assigned to one or more repository groups. When a new username is assigned to a group, the user is granted every privilege granted to the group. A username can be assigned to one of the default repository groups. However, for increased repository security, custom groups should be created with task-appropriate privileges.

A user's group affiliation can be changed at any time. When a user's group is changed, the user is granted the privileges of the new group.

## **Users and Privileges**

When the administrator assigns a user to a user group, the user receives all privileges granted to the group. Privileges can also be assigned to users individually. When a privilege is granted to an individual user, the user retains that privilege even if the user changes group affiliation.

For tighter security, grant the Super User privilege to the individual user, not the entire Developer group. This limits the number of users with the Super User privilege, and ensures that the user retains the privilege even if the user is removed from the Developer group.

## **Users and Folders**

Any user can be the owner of a folder in the repository. You then grant separate levels of privileges to the user, and to one of the user's groups.

### **11.2.5. Editing a User Password**

You can edit a user password without editing the user properties. You can edit your own password if you have the Browse Repository privilege. You can edit the password of every user in the repository if you have one of the following privileges:

- Administer Repository
- Super User

### **11.2.6. Repository Privileges**

The Repository Manager grants a default set of privileges to each new user and group for working within the repository. You can add or remove privileges from any user or group except:

- Administrators and Public
- Administrators and the database user who created the repository

## **Privileges and Permissions**

You can perform some tasks in the repository with only repository privileges, such as stopping the Informatica Server or creating user groups. Folder-related tasks, however, generally require one or more folder permissions in addition to the related repository privilege to perform the task.

### **Folder Permissions**

The available folder permissions are:

- Read permission. Allows users to view the folder as well as objects in the folder.
- Write permission. Allows users to create or edit objects in the folder.
- Execute permission. Allows users to execute or schedule a session or batch in the folder.

Developers can edit folder permissions and properties at any time. To edit folder properties, the developer must have one of the following:

- Browse Repository privilege, as the folder owner with read permission
- Administer Repository privilege with read permission
- Super User privilege

### Repository Locks

The repository uses locks to prevent users from duplicating or overriding work. It will allow multiple users to obtain read locks on an object to view. It will allow only one write lock per object. This keeps multiple users from editing the object at one time, thus preventing repository inconsistencies. If a user attempts to edit an object that already has a write lock, the repository displays a message box:

The [object\_type] [object\_name] is already locked by [username].

The repository then issues a read lock for the object, allowing the user to view the object.

The repository allows only one execute lock per object. This keeps a user from starting a session that is already running, which can cause the Informatica Server to load duplicate or inaccurate data to the data warehouse.

Table 12 – Repository Locks

Repository Lock	Created When	Max per object
Read	Viewing an object in a folder for which the user does not have write permission or attempting to view an object that is already write-locked.	Unlimited
Write	Editing a repository object in a folder for which the user has write permission.	1
Execute	Starting a session or batch or when the Informatica Server starts a session or batch as scheduled.	1
Fetch	Accessing information from the repository.	Unlimited
Save	Saving the information to the repository.	1

### 11.2.7. Locking Within Objects

Some repository objects contain other repository objects. For example, sessions contain mappings, and mappings contain at least one source and target definition. If you save changes to an object used by other objects, the repository might make the other objects invalid. Before you can use invalidated objects, you must validate them. To validate a session, you must open the session property sheet and save it.

### 11.2.8. Locking with Cubes and Dimensions

Editing or deleting cubes and dimensions can affect many objects in the repository. When you edit a property of a cube or dimension, the Designer creates a write lock on all related

objects until you save your changes or cancel your edit. Therefore you might notice an object is locked even when no one is working with it if that object is a part of a cube or dimension being edited.

### **11.2.9. Locking Business Components**

To maintain the integrity of your repository data, the Designer locks the business component tree while its contents are being edited, preventing you from copying or editing the business component. Locking occurs at the root directory of the business component tree.

### **11.2.10. Handling Locks**

Sometimes, the repository does not release a lock. This can happen when:

- Network problems occur.
- An Informatica Client, Informatica Server, repository, or database machine shuts down improperly.

If an object, version, or folder is locked when one of these events occurs, the repository does not release the lock. This is called a residual lock. If you are the owner of the residual lock, you can sometimes reacquire the lock when you continue work with the object. If you are not the owner, you can use the Repository Manager to determine who owns the lock, and then unlocks the object, version, or folder.

### **11.2.11. Viewing a Lock**

You can view existing locks in the repository in the Repository Manager. The Repository Manager provides two ways to view locks:

- **Browse the repository.** Use the Navigator and main windows to display the folders, versions, and objects in use.
- **Show locks.** Use a menu command to view all locks in the repository. This method provides more detailed information and allows you to sort your view of the locks.

Since users can save and close objects at any time, when using either method, refresh your view of the repository for the most accurate lock information.

### **11.2.12. Tips**

When setting up data warehouse security, keep it simple. You have the tools to create a complex web of security, but the simpler the configuration, the easier it is to maintain. Securing the environment involves three basic principles:

- Limit users
- Restrict privileges
- Define permissions

### **11.2.13. Create groups with limited privileges**

The simplest way to prevent security breaches is to limit the number of users accessing the repository. Then, limit the ability of other repository users to perform unnecessary actions, such as running sessions or administering the repository.

To do this, determine how many types of users access the repository. Then create separate user groups for each type. The more distinct your user groups, the tighter your data warehouse security. Once you establish separate groups, assign the appropriate privileges to those groups and design folder permissions to allow their access to particular folders.

- Do not use shared accounts.

Using shared accounts negates the usefulness of the repository lock feature. The repository creates locks on objects in use. Breaking a valid lock can cause repository and data warehouse inconsistencies.

- Limit user and group access to multiple repositories.

When working in a multiple repository environment (PowerCenter only), limit the number of users and groups accessing more than one repository. This becomes more important when repositories contain folders with the same or similar names. Restricting the number of crossover users limits the possibility of a user connecting to the wrong repository and editing objects in the wrong folder.

- Customize user privileges.

If a single user requires more privileges than those assigned to the user's group, and you need to keep the user in that group for the purpose of folder permissions, you can add individual privileges to that user.

- Limit the Super User privilege.

The Super User privilege permits you to perform any task despite folder-level restrictions. This includes starting any session or batch, and unlocking other user's locks. To protect your repository and data warehouse, restrict the number of users who need this all-encompassing privilege.

- Limit the Administer Repository privilege.

The Administer Repository privilege permits a user to copy a folder from one repository to another. Since this feature is often used to deploy folders into production, you should limit this privilege in production repositories.

- Restrict the Session Operator privilege.

With the Session Operator privilege, you can use the Server Manager to start any session or batch within the repository for which you have read permission. You can also use the command line program pmcmd to start any session or batch in the repository. Misuse of this privilege can result in invalid data in your data warehouse.

When possible, avoid granting the Session Operator privilege. Instead, if the user needs to use the Server Manager to start sessions or batches, you can assign the user the Create

Sessions and Batches privilege, and read and execute permission for the folders in which the user works. If the user uses pmcmd to start sessions or batches, the user needs only execute permission for the folder.

### **11.3. Data Warehouse RDBMS**

The following general database security techniques can be used:

- Security views
- Partitioned fact tables
- Split fact tables

#### **11.3.1. Security Views**

Most databases provide a way to restrict access to data. For example, a user may be able to access only certain tables or he may be restricted to certain rows and columns within a table. The subset of data available to a user is called the user's security view.

Note that restrictions on tables, or rows and columns within tables, may not be directly evident to a user. They do, however, affect the values displayed in a report. You need to inform users as to which data they have access so that they do not inadvertently run a report that yields misleading final results. For example, if a user only has access to half of the sales data in the warehouse but runs a summary report on all sales, the summary will only reflect half the sales. Reports do not indicate the database security view used to generate the report.

Consult your database vendor's product documentation to learn how to create security views for your particular database.

#### **11.3.2. Partitioned fact tables**

You can partition a fact table in order to group rows together. The resultant partitioned tables are physically distinct tables in the data warehouse and security administration is easy because permissions are granted to entire tables rather than rows and columns.

Partitioned fact tables are invisible to system users. Although there are many physical tables, the system "sees" one logical fact table. Support for partitioned fact tables for security reasons should not be confused with the support that MicroStrategy Intelligence Server provides for partitioned fact tables for performance benefits.

#### **11.3.3. Split fact tables**

Split fact tables are the creation of two or more physical fact tables from one logical fact table. Each new table contains the same primary key, but is a subset of the fact columns of the original fact table. Splitting fact tables allows fact columns to be grouped based on user community. This makes security administration easy because permissions are granted to entire tables rather than to columns.

## 11.4 MicroStrategy Security Features

In general, security systems have the following components:

- **Authentication:** A way to identify yourself to the system
- **Access Control:** What users are allowed to see and do once they have identified themselves
- **Auditing:** A record of what users saw and did

### 11.4.1. MicroStrategy Application Security

The following is a list of the different types of authentication modes in the MicroStrategy environment:

- Standard
- Windows NT
- Anonymous
- Database

#### **Standard authentication**

Standard authentication allows users to identify themselves using a system login ID and password. The system login ID is unique across the entire system. When a project source is configured to use standard authentication, users must enter a valid login ID and password combination before they can access the project source.

#### **Windows NT authentication**

Windows NT assigns a unique security identifier (SID) to every user in the NT network. With Windows NT authentication, users are identified by their Windows NT SID and they are not prompted to enter a login ID and password.

To allow Windows NT authentication, the developer must link the users in the MicroStrategy environment to Windows NT users. Linking allows MicroStrategy Intelligence Server to map a Windows NT user to a MicroStrategy user.

#### **Anonymous authentication**

Anonymous authentication allows users to access the system as a Guest user with a minimum set of privileges. Guest users inherit their privileges from the Public group and they are not part of the Everyone group.

To allow anonymous access to a server, the developer must grant connect access to the Public group. To allow anonymous access to a project, the developer must include the Public group in one of the project's security roles.

## **Database authentication**

Database authentication identifies users using a login ID and password for the metadata and warehouse databases. This can only be used with projects created in the MicroStrategy 6 Suite.

When a project source is configured to use database authentication, the user is prompted for a login ID and password combination for both databases. MicroStrategy Intelligence Server passes the login information to the databases and the databases determine whether or not the information is valid.

### **11.4.2. MicroStrategy 7.0 Access Control**

Access control determines what users are allowed to see and do once users have identified themselves on the system. There are two types of access control in the MicroStrategy environment.

- Privileges
- Permissions

The MicroStrategy Suite provides the following security services to implement access control:

User and group administration

- Security roles
- Security filters
- Connection mapping

#### **Privileges**

Privileges define the types of actions that particular users and groups may perform in the system. There are three types of privileges:

- Object creation privileges – specify the types of objects a user may create
- Application access privileges – specify the editors, dialogs, and wizards with which a user may interact
- System privileges – system-wide privileges such as whether a user is allowed to backup the system, take ownership of an object, log another user of the system, and so on. These privileges are independent of a specific project.

#### **Permissions**

Permissions define which users and groups have access to what objects and the degree to which they can access those objects.

MicroStrategy Intelligence Server 7.0 users the following to enforce permissions for an object:

- The authenticated user attempting to access that object
- The owner of the object

- The access control list of the object

### **Authenticated user**

The authenticated user provides the following information to MicroStrategy Intelligence Server 7.0:

- User identity – Determines an object's owner. Also determines whether or not a user has been granted the right to access an object.
- Group membership – A user is granted access to an object if he belongs to a group that has access to the object.
- Special privileges – A user may possess a special privilege that causes the normal access checks to be bypassed.

### **Object owner**

Objects keep a record of their current owner. Typically, the owner is the user who created the object. The owner or an administrator decides who may access the object and what type of access is granted.

### *Access control list*

The access control list of an object is a list of users and groups and the permissions that each one has for the particular object. Access control lists have the following information:

- User: The name of the user or group that is granted or denied access to the object.
- Permissions: The degree to which the user or group is granted or denied access to the Object: The available permissions are:
  - **Browse** – allows users to see an object in the folder list and object viewer
  - **Use / Execute** – allows users to use an object needed for execution for example, a filter that needs to be used in a report execution
  - **Read** – allows users to view the object's definition and access control list
  - **Write** – allows users to modify the object definition, but not the object's access control list
  - **Delete** – allows users to delete the object
  - **Control** – allows users to modify the access control list of an object and take ownership of an object
- Inheritable: Applies only to folders. If set, any object placed in the folder will inherit the folder's entry in the access control list.

## User and group administration

MicroStrategy Intelligence Server allows administrators to create, modify, and delete users and groups. You can assign privileges to or revoke privileges from individual users or entire groups of users.

The following groups are provided by default:

- Everyone

All users except for guest users are automatically members of the Everyone group. The Everyone group is provided to make it easy for administrators to assign privileges, security role memberships, and so on to everyone in the system.

- Public

The Public group provides the capability for anonymous logins and is used to manage the access rights of guest users. If the administrator chooses to allow anonymous authentication, each guest user assumes the profile defined by the Public group. When a user logs in as a guest, a new user is created dynamically and becomes a member of the Public group.

- System Monitors

The System Monitors group provides an easy way to give users basic administrative privileges in the system. Users in the System Monitors group have access to all of the monitoring tools under the Administration section of a project source's folder list. However, System Monitors cannot modify any configuration objects such as database instances, server configurations, governors, and so on.

- System Administrators

The System Administrators group is a group within the System Monitors group. It provides all the capabilities of the System Monitors group plus the ability to modify all system objects.

- Web SE Users

The Web SE (Standard Edition) Users group provides an easy way to give users access to MicroStrategy Web functionality. The Web SE Users group is assigned privileges associated with standard Web functionality.

- Web PE Users

The Web PE (Professional Edition) Users group provides an easy way to give users access to advanced MicroStrategy Web functionality. The Web PE Users group is assigned privileges associated with advanced Web functionality. The Web PE Users group is a group within the Web SE Users group; it provides all the privileges of the Web SE Users group plus additional privileges.

- Security Roles

Security roles are collections of privileges that can be reused from project to project. For example, an administrator may create a security role that allows users to access all the

editors except for the Document Editor. Once the administrator creates this security role, it can be saved and used in any project registered with the server. The users associated with a particular security role can vary by project.

The following security roles are provided by default:

- *Normal users*: no privileges are granted
- *Power users*: all privileges are granted

### **Security filters**

Security filters prevent users from seeing certain data in the database. If two users with different security filters run the exact same report, they may get different results. For example, a regional manager may have a security filter that only allows her to view data from her particular region regardless of the report she runs.

A security filter has the following parts:

- *Filter expression* – specifies the subset of the data that a user can analyze
- *Top range attribute* – specifies the highest level of analysis to which the security filter is applied.
- *Bottom range attribute* – specifies that lowest level of analysis to which this security filter is applied.

### **Connection mapping**

By default, all users use the same database connection and the same database login when submitting queries to the warehouse database. Connection mapping allows administrators to map particular users to different database connections and different database logins.

## 12 Data Warehouse Performance Considerations

### 12.1. Introduction

This section outlines the basic database tuning methodology for decision support systems. It lists the objectives, techniques and procedures to execute the first phase of tuning a database for decision support. This is not a comprehensive. Application tuning should be implemented.

Proper database tuning has the following purposes:

- High performance for frequent queries
- Acceptable performance for uncommon or complex queries

The typical process for database tuning has the following steps:

- 1) Obtain reporting requirements
- 2) Perform preliminary tuning
- 3) Deploy and monitor
- 4) Fine-tune

This section focuses on item 2.

### 12.2. Database Tuning Techniques

Basic tuning techniques include

- Aggregate Tables
- Indexes
- Denormalization
- Partitioning

#### 12.2.1. Aggregate tables

Aggregate tables store pre-summarized totals at a lever higher level of aggregation than the most granular fact table. They allow reports to be generated from small rather than large tables. A successful aggregation strategy seeks to choose aggregate tables that will have the most impact while taking the least amount of space.

Aggregation decisions are driven by the following factors:

- *Usage patterns.* Aggregate tables that are likely to be used the most should be built and maintained.

- *Compression ratios.* The compression ratio between two tables is defined as the size of the aggregate compared to the size of the smallest table that the aggregate can be derived from
- *Volatility.* Changes in hierarchies over time will impact the accuracy of aggregate tables. Sometimes aggregate tables need to be rebuilt as a result of changes in dimensions

Some rules of thumb include

- Aggregation typically yields about 80% of the performance improvement achieved with tuning.
- A good candidate for aggregation should have at most 10%-15% of the size of the smallest table from which it can be derived.

### **12.2.2. Indices**

Indices are data structures that store the physical location of all rows matching each value of the index key (the column or columns being indexed). Indices are designed for fast access to specific values of the index key. Indices are useful for highly selective queries, that is, for queries that will return a small percentage of the rows in the table being queried. An example of a highly selective query is a request for the records for a specific client from a sales table that contains data for 1000 clients. Drilling a few times will typically generate a highly selective query. Consider the range of a column when designing indices. The range of a column is the set of different values that the column value can take. The higher the cardinality of the range, the better candidate the column is for a b-tree index. For example, an index on "client" (1000 different values) will have a lot more impact than an index on "year" (3 different values). Bitmap indexes are effective for low cardinality values. In the previous, a bitmap index should be created for "year".

Rule of thumb. Do not build indices on tables with less than 100 rows. (Take this number with a grain of salt; a better measure would be the number of blocks, which will vary with the size of each record)

### **12.2.3. Denormalization**

Denormalization means the introduction of redundancy, typically by adding extra columns to existing tables. Denormalization improves performance by reducing the number of joins. Denormalization of lookup tables is a standard recommendation. Denormalization of fact tables is also supported. This technique will have high maintenance costs for dimensions that frequently change over time. This occurs because changes in a dimension would require updates to the fact table.

### **12.2.4. Partitioning**

Today's data warehouses contain vast arrays of data. Most of this data will be held in a few very large fact tables. When user's queries hit these tables, they are requesting a minuscule percentage of all the data in these tables. If these large "base" tables can be divided up or

partitioned, the query can search a smaller subset; this advantage will allow the query to return much faster.

Table partitioning is a technique by which a very tall table is made shorter by splitting it into several smaller tables. This allows queries to be targeted at a specific range of data, often yielding substantial performance improvements. However, if multiple ranges of data need to be scanned, then partitioning can result in poor performance because multiple partitions will need to be scanned. Table partitioning creates virtual or actual tables that are subsets of larger tables.

For example, imagine a table that contains the Sales for Items in Stores. There are two years of data in this table, at the Week level. Partitioning by Week would create 104 Tables (52 Weeks \* 2 Years). Each partition only holds data for a single week. Users may query single partitions. If users ask for a ranking of Last Week's Sales, they could scan only one partitioned table rather than having to run against the base table. Thus they would be utilizing a table with 1/104 the data. Similar to the strategies for pre-aggregating, partitioning ought to be based on usage patterns. Typically, the SQL that is being created by the users will specify a hierarchical level in the WHERE clause. This level is the first to consider in developing a partitioning strategy.

Partitioning can happen at the application level or at the server level. Server-level partitioning is executed by some RDBMS software packages. This creates virtual tables in the database. Creation and maintenance of these tables are handled by the RDBMS. MicroStrategy takes care of application-level partitioning. Our metadata stores the partitioning strategy and our SQL-parsing engine will use this to force queries to only hit the smaller partitions.

### **12.2.5. Preliminary Database Tuning Methodology**

#### **Know your model first**

Areas of interest include, from most too least important

- Cardinality of each attribute
- Most frequently queried attributes
- Estimations of compression factors
- Attributes that are frequently queried together
- Attributes that are rarely queried together
- Volatility of each dimension

Attributes that are queried together should be kept together in aggregate tables. Attributes that are not queried together can be aggregated independently.

#### **Preliminary tuning for Oracle**

- Create a primary key on the id column for each lookup table.

- Create single column bitmap indexes on all foreign keys (lowest level attribute for each dimension) in the fact and aggregate tables.
- Create a composite primary key made up of each foreign key (as above) in all fact and aggregate tables. Order of columns within Primary Key should have most commonly accessed columns in terms of the WHERE clause first.
- Generate statistics on all tables
- Make note of and fix the following parameters in the init.ora file:
  - BITMAP\_MERGE\_AREA\_SIZE (should be at least 1 MB)
  - FAST\_FULL\_SCAN\_ENABLED (Make sure this is true! Its default is false)
  - HASH\_AREA\_SIZE (document this)
  - SHARED\_POOL\_SIZE (document this)
  - SORT\_AREA\_RETAINED\_SIZE (document this) SORT\_AREA\_SIZE (document this)
  - HASH\_JOIN\_ENABLED (should be TRUE)
  - OPTIMIZER\_MODE (should be CHOOSE)
  - STAR\_TRANSFORMATION\_ENABLED (need to make sure this is TRUE, this is false by default)

### **12.3. MicroStrategy Intelligence Server Thread Management**

MicroStrategy Intelligence Server manages the number of user connections to the data warehouse. Database connections must be managed between two processes:

- User requests and the size of report request queues
- The performance of the data warehouse

As more connections are opened to the data warehouse, a greater number of requests are made to the data warehouse at any one point in time. Additional load on a RDBMS will typically degrade performance. Therefore, it is recommended to manage the number of database connections between MicroStrategy Intelligence Server and the data warehouse so that report throughput (the number of reports generated within a given period of time) is maximized. Determining the correct number of database connections can only be done through testing and benchmarking.

### **12.4. Informatica PowerCenter**

Tuning PowerCenter involves the ability to enable sessions to run at optimal speed. To help determine where the session performance can be improved, the Informatica Server can create a set of information known as “performance details.” Performance details provide transformation-by-transformation information on the flow of data through the session. With

session performance details, developers can improve the overall performance of your session.

You can view performance details through the Server Manager as the session runs or after the session completes, or you can open the resulting file in a text editor. You create performance details by selecting Perform monitor in the session property sheet before running the session. By evaluating the final performance details, you can determine where session performance slows down, in the source or target databases, or in the server. Monitoring also provides session-specific details that can help tune:

- Buffer block size.
- Index and data cache size for Aggregator, Rank, and Joiner transformations.
- Lookup transformations.
- Before using performance details to improve session performance you must:
- Enable monitoring.
- Understand counters.

### **Enabling Monitoring**

To view performance details, you must enable monitoring in the session property sheet before running it.

To enable monitoring:

- 1) In the Server Manager, open the selected session property sheet.
- 2) On the General tab, select Collect Performance Data, and click OK.
- 3) Run the session.

### **Viewing Session Performance Details**

You can view session performance details through the Server Manager or by locating and opening the performance details file.

In the Server Manager, you can watch performance details during the session run.

*To view performance details in the Server Manager:*

- 1) Monitor the server.
- 2) Select the session in the Monitor window and click the Session Performance Details button, or choose Server Requests-Session Performance Details.
- 3) If you want to watch the information as it changes, click Refresh Continuously.

*To view the performance details file:*

- 1) Locate the performance details file.

The server names the file `session_name.PERF`, and stores it in the same directory as the session log. If there is no session-specific directory for the session log, the server saves the file in the default log files directory.

- 2) Open the file in any text editor.

Often performance slows because your system relies on inefficient connections or an overloaded Informatica Server system. By making the appropriate global changes, you can improve the performance of all your sessions.

### Accessing Data over a Network

Server performance is directly related to network connections. Data generally moves across a network at less than 1 MB per second, whereas a local disk moves data five to twenty times faster. Thus, the fewer network hops between the Informatica Server, and source and target databases, the faster the Informatica Server processes sessions. Consider the following options for minimizing hops to improve server performance:

- **Flat files.** When your flat files are separated from the Informatica Server, server performance becomes dependent on the performance of your network connections. Moving the files onto the server system, adding disk space, if necessary, will improve performance.
- **Relational databases.** Where possible, minimize the number of network hops between the source and target databases and the Informatica Server. Moving the target database onto a server system may improve server performance.
- **Staging areas.** If you use a staging area, you force the Informatica Server to perform multiple passes on your data. Where possible, remove staging areas to improve performance. The server can read multiple sources with a single pass, which may alleviate your need for staging areas.

### Using Multiple Informatica Servers

You can run multiple PowerCenter servers on separate systems against the same repository. If you have multiple PowerCenter servers, distributing the session load to separate server systems increases performance.

### Eliminating Paging

Paging occurs when the Informatica Server system runs out of memory for a particular operation and uses the local disk for memory. Use system tools to monitor paging activity. Since paging slows performance considerably, where possible, increase the server system memory allocation to reduce paging. If this is not possible, you may want to add memory to your system.

Many factors can affect the performance of your sessions: database efficiency, buffer size, and transformation choices, to name a few. Some are easy to spot, others are more difficult. No matter which techniques you use, your goal should be to tune your sessions to run during the available load window, the time you have to use the Informatica Server. For example,

most database administrators schedule large or complicated sessions to run at night, providing the Informatica Server with several uninterrupted hours to process data for the next day. As long as the scheduled sessions run within that load window, there is no pressing need to tune those sessions. If, however, you have sessions that run into the morning, or if they need to run during the day on a regular basis, those sessions might benefit from judicious tuning.

When tuning sessions, you find and tune the worst bottleneck first. Once you remove this bottleneck, the second-worst bottleneck becomes the worst, and so on. Continue tuning until the session runs within the given load window.

You can tune performance in four general areas:

- **System.** To increase performance, first tune your system. Tuning your system setup, such as decreasing the number of network hops between the Informatica Server and databases or adding multiple Informatica Server on separate systems, can double or triple the performance of your sessions.
- **Source and target databases.** Use performance details to locate bottlenecks in the source or target databases. Then have the administrator optimize database performance.
- **Session.** Once you optimize the databases, you can focus on the session. You can optimize the session strategy and use performance details to help tune session configuration. By tuning the session, along with the source and target databases, you can increase session performance by 20-50 percent.
- **Mapping.** To potentially increase performance another 10 to 20 percent, you can fine-tune the mapping, streamlining expressions and trimming transformations.

Because determining the best way to improve performance can be complex, change only one variable at a time, and time the session both before and after the change. If session performance does not improve, you may want to return to your original settings.

## 13 Data Warehouse Operations Architecture

### 13.1 Introduction

Maintaining the DWA will require several tools to monitor each of the system components. These tools can be broken into Informatica, Data Warehouse, and MicroStrategy tools.

### 13.2 Informatica

The PowerCenter Server Manager is the application that enables the user / developer to create, schedule, monitor, and performance tune sessions. Either the Server Manager or the command line program "*pmcmd*" is used to start a session. A session can be configured to run on demand, or to run on a set schedule.

### 13.3 Data Warehouse

Database monitoring tools will be required for the production system. These tools will be used to track system usage, user connections, and manage database objects. In order to maintain the performance of the RDBMS, monitoring with these tools is needed.

### 13.4 MicroStrategy

MicroStrategy Administrator provides a Warehouse Monitor application that can monitor and track the various projects and application built on the MicroStrategy Platform. Warehouse Monitor allows administrators to monitor and track user connections, project usage patterns, report usage patterns, report object cache, and reporting queues. With Warehouse Monitor, administrator can maintain and often improve the overall system performance.

MicroStrategy Administrator also allows for the transfer of user reporting objects. Reporting objects owned by one user can be copied or moved so that they are accessible to user groups or to all users.

## 14 Data Warehouse Administrative Specifications

### 14.1. Sun Solaris Administration Procedures

#### 14.1.1. Informatica PowerCenter Server Start-up Procedures

The PowerCenter server will run on the Sun Solaris platform. The server can be started from the actual Sun server or from a remote machine (via Telnet, etc.). To start the PowerCenter server, follow these procedures:

From the Unix command line:

- 1) Verify that the repository database is running.
- 2) Connect to the Unix machine on which the server is running. Logon as pmrepo user with PowerCenter password.
- 3) Change the directory to the /Informatica/PowerCenter directory (where the pmserver.cfg file is located).
- 4) Type pmserver. (Will say "Server starting up...ok." Ignore this message and proceed).
- 5) Type ps - ef | grep pmserver
- 6) Two lines should be displayed if server is running. If only one line is returned, open the pmserver.log file and respond to error messages as appropriate.

#### 14.1.2. Informatica PowerCenter Server Shutdown Procedures

The PowerCenter server is stopped using the Server Manager module from the client machine. To stop the PowerCenter server, follow these procedures:

From the Server Manager:

- 1) Log on to the Server Manager module with administrative privileges.
- 2) Select the server icon that is to be stopped.
- 3) From the Server Requests option, select Stop Server.
- 4) Review messages in the output window to confirm successful stop.

#### 14.1.3. Sun Solaris Backup Procedures

Configuration information for the Solaris server is contained in the pmserver.cfg file, located in the /Informatica/PowerCenter directory. This file should be backed up and stored separately from the server in case the server needs to be reconfigured. The file contains license keys, connection information, as well as the other parameters set at configuration.

Repository information should also be backed up at least once daily using the Repository Manager. To perform a backup, do the following:

- 1) Sign on to the Repository Manager as an Administrator.
- 2) From the Repository Menu, select the Backup Repository option and select file location.

#### **14.1.4. Sun Solaris Recovery Procedures**

On Unix, the PowerCenter Server creates a log for all status and error messages named `pmserver.log`. An error log for error messages only is contained in the `pmserver.err` file. These files, located in the `pmrepo/Informatica/PowerCenter` directory, can be used to troubleshooting and recovery of failed sessions or server services.

#### **14.1.5. Sun Solaris Performance and Tuning**

The goal of performance tuning is optimize session performance so sessions run during the available load window for the Informatica Server. The following are various factors that can affect system performance:

- 1) Hard disk speed on related machines.
- 2) Network speed.
- 3) CPUs on related machines.
- 4) Configure physical memory for the Informatica Server to minimize disk I/O.

When the Informatica Server runs out of physical (RAM) memory, it starts paging to disk to free physical memory. This will slow performance, so the Informatica Server should be equipped with sufficient RAM (2 Gig or more is recommended) to minimize paging to disk.

#### **Optimize the database configuration**

The database administrator should ensure that the source and target databases and the repository have been optimized.

Performance will generally suffer due to inefficient connections or an overloaded server. By making the appropriate global changes, overall performance for the ETL process can be improved.

#### **Accessing Data Over a Network**

Server performance is directly related to network connections. The fewer connections between the Informatica Server, and source and target databases, the faster the server can process the sessions. The following can minimize these connections:

- **Flat files.** When flat files are stored on a separate machine from the server, session performance becomes dependent on network connections. Moving files onto to the server and sourcing them locally will improve performance.
- **Relational databases.** When possible, the number of network connections between the source and target databases and the Informatica Server should be minimized. Thus, the optimal location for the data warehouse is the Informatica Server.

- **Staging areas.** If a staging area is used, the Server must perform multiple passes at the data. Also, multiple sources can be read during the extraction process to eliminate a staging area where possible.
- **Multiple Servers.** PowerCenter provides the capability to run multiple servers on separate systems against the same repository, improving overall performance.

### **Creating Performance Details**

To help determine where the session performance can be improved, the Informatica Server can create a set of information known as performance details. Performance details provide low-level information on the flow of data through a given session that can be used to optimize the system. This information can be viewed through the Server Manager as the session runs or after the session completes.

To create Performance Details, select “Perform Monitor” in the session property sheet before running the session. By evaluating these details, it is possible to tell where performance slows down, in the source or target databases, or in the server. Monitoring also provides session-specific details that can help tune the following:

- Buffer block size.
- Index and data cache size for Aggregator, Rank, and Joiner transformations.
- Lookup transformations.

## **14.2. NT Administration Procedures**

### **14.2.1. General Startup Procedures for MicroStrategy products**

MicroStrategy server products run as a Windows NT service. This allows users to start up and shut down servers from a remote machine. In addition, it is recommended that a developer configure the service to start automatically when the machine on which it is running starts up.

When shutting down the data warehouse or MicroStrategy Intelligence Server, the following products should be restarted in the following order:

- 1) Data warehouse RDBMS and MicroStrategy metadata repository RDBMS
- 2) MicroStrategy Intelligence Server
- 3) MicroStrategy Web
- 4) MicroStrategy Desktop, Agent, Architect, Administrator

### **14.2.2. MicroStrategy Intelligence Server Startup Procedures**

If pending jobs exist in a queue when MicroStrategy Intelligence Server was last shut down with the “Shut down Server” option, it is restored when MicroStrategy Intelligence Server is started. Old jobs continue to execute and new jobs are accepted. If MicroStrategy

Intelligence Server is not shut down properly, then jobs in a queue will be lost and only new jobs will be executed.

### **14.2.3. MicroStrategy Intelligence Server Shut down Procedures**

When MicroStrategy Intelligence Server shuts down, all projects are idled immediately. It stops accepting new client requests and cancels all currently executing jobs. These canceled jobs are saved and will be executed when MicroStrategy Intelligence Server starts up again.

During the shutdown process the MicroStrategy Intelligence Server service is stopped and all projects are unloaded from the server. The current server state is captured and saved to disk so it can be restored when the server starts again.

### **14.2.4. MicroStrategy Intelligence Server Backup**

State information is information about the current operating state of MicroStrategy Intelligence Server. MicroStrategy Intelligence Server state information includes:

- Run time configuration
- Any open jobs and their corresponding users
- Any projects that are loaded

All state information is backed up periodically, so that the current state can be restored the next time MicroStrategy Intelligence Server is started. MicroStrategy Intelligence Server stores backups of state information in:

- *Metadata/registry*. All run time configuration parameters are stored in the machine's registry and in the metadata.
- *Transaction log*. Records a history of client transactions with MicroStrategy Intelligence Server that required the creation of a job.
- *Snapshot backup*. Represents the state of MicroStrategy Intelligence Server frozen at a particular point in time.

### **14.2.5. General NT Performance and Tuning Recommendations**

The following steps are recommended in order to maintain the NT Server operating system on the NT machines that hold MicroStrategy products.

#### **Disk fragmentation**

The disk(s) fragmentation should be checked at least once a month. De-fragment the hard disks whenever a bad fragmentation is observed. Indeed, a fragmented system will take longer to access and create files. These processes will require several disk accesses instead of one. Files will become spread across the disk and will no longer be contiguous. Diskkeeper Lite is a commonly used tool.

### **Temp folder size (typically ... \Temp)**

Many applications, including MicroStrategy applications, save by default in this folder some temporary files needed for proper operation. The “write” actions of these applications in their respective temporary files could potentially be slowed down if the Temp folder is big. In case it is full, these applications could even be unable to complete this "save" action and won't work properly. Regularly empty the Temp folder or delete some files to prevent this from happening.

### **Log file size**

Some log files are enabled by default. Their size should be controlled since the logging could be slowed down if they are too big. Periodically check the size of these files and clean them. Check MicroStrategy Product manuals for details about the log files.

In addition, be aware of the fact that any logging affects performance. Under normal operation conditions, only minimum logging should be enabled. Whenever extra logging is enabled – ODBC Trace, MicroStrategy Server logs, etc..., for troubleshooting reasons for example, make sure it is disabled after collection of the data needed.

### **MicroStrategy Server Machines Cleanliness**

On the production MicroStrategy Server machines, only the MicroStrategy product instances being used in production should be kept. Delete – or at least disable in the Control Panel Services window - any other instances that could have been created.

Optimize the use of hard disk space: the set-up of different drives for a separate storage of the Operating System files and the “Application” files will improve the machine processing performance.

A "Cold Boot" of the MicroStrategy NT server machines should be performed once a month to clean up potential memory-related issues (leaks from the ODBC drivers, etc.). Take as well the good habit to re-start regularly the Client machines (especially the Project Administrator's and the Power User's ones): this will clean up the machine memory. This will ensure that the memory resources are fully available to be used by the project applications.

## 15 Glossary of Terms and Acronyms

### 15.1. Terms

Table 13 – List of Terms

Term	Definition
Ad Hoc Query	Any query that cannot be determined prior to the moment the query is issued.
Architecture	A definition and preliminary design which describes the components of a solution and their interactions. Architecture is the blueprint by which implementers construct a solution that meets the users' needs.
Data Administration (DA)	The processes and procedures by which the integrity and currency of the data in the warehouse are maintained.
Data Aggregation	A type of data derivation where a data value is derived from the aggregation of different data occurrences of the same subject data. For example, yearly sales data can be aggregated from monthly sales data.
Data Cleansing	The process of removing errors and resolving inconsistencies in source data before loading the data into a target environment
Data Dictionary	It is a collection of definitions and specifications for data categories and their relationships. It is a database of data about data (metadata).
Data Extract	The process of copying a subset of data from a source to a target environment.
Data Mart	A type of data warehouse designed to meet the needs of a specific group of users such as a single department or part of an organization. Typically a data mart focuses on a single subject area such as sales data. Data marts may or may not be designed to fit into a broader enterprise data warehouse design.
Data Model	A logical map that represents the inherent properties of the data independent of software, hardware or machine performance considerations. The model shows data elements grouped into records, as well as the associations between those records.
Data Propagation / Replication	A process for distributing data from a source database to target data-bases while usually keeping the databases synchronized used and presents a consistent and commonly understood and accepted view and definition of the enterprise data.
Database	A collection of data which are logically related.
Database Management Systems (DBMS)	A software system for creating, maintaining and protecting databases.
Data Scrubbing / Transformation	The process of filtering, merging, decoding, and translating source data to create validated data for the data warehouse. For example, a numeric regional code might be replaced with the name of the region.
Data Warehouse	A subject oriented integrated, time-variant, non-volatile collection of data in support of management's decision making process. A repository of consistent historical data that can be easily accessed and manipulated for decision support.
Decision Support System (DSS)	Systems that allow decision makers in organizations to access data relevant to the decisions they are required to make.

<b>Term</b>	<b>Definition</b>
Derived Data	Warehouse data that results from calculations or processing applied to the source data before it is stored in the Warehouse environment. For example, source data might be used to calculate ROI (Return On Investment) which is stored as derived data in the Warehouse.
Drill Down	A method of exploring detailed data that was used in creating a summary level of data. Drill down levels depend on the granularity of the data in the data warehouse.
Enterprise Data Warehouse	An Enterprise Data Warehouse is a Centralized Warehouse that services the entire enterprise. Enterprise Data Warehouses are sometimes used to populate data marts.
Executive Information System (EIS)	Tools programmed to provide canned reports or briefing books to top-level executives. They offer strong reporting and drill-down capabilities. Today these tools allow ad-hoc querying against a multi-dimensional view of data, and most offer analytical applications along functional lines such as sales or financial analysis.
Extract	The process of copying a subset of data from a source to a target environment.
Informational Applications	Applications which are written to analyze data from a Data Warehouse for Decision Support purposes.
Enhanced Data	Warehouse data that has been cleansed, scrubbed, transformed, derived, summarized, or aggregated.
Enterprise Data Model	A blueprint for all of the data used by all departments in the enterprise. An Enterprise Data Model has resolved all of the potential inconsistencies and parochial interpretations of the data
Metadata	Data about data. For example, information about where the data is stored, who is responsible for maintaining the data, how often the data is refreshed, etc.
Middleware	A communications layer that allows applications to interact across hardware and network environments.
Multi-Dimensional Analysis (MDA)	Informational Analysis on data taking into account many different relationships, each of which represents a dimension. For example, a person doing an analysis of retail may want to understand the relationships among sales by region, by quarter, by demographic distribution (income, education level, gender, or by product). Multi-Dimensional Analysis will yield results for these complex relationships. Multi-Dimensional Analysis is sometimes referred to as On-line Analytical Processing or OLAP.
On-Line Analytical Processing (OLAP)	Processing that supports the analysis of business trends and projections. It is also known as Multi-Dimensional Analysis.
Operational Applications	Applications which support the daily operations of the enterprise. Usually included in this class of applications are Order Entry, Accounts Payable, Accounts Receivable, etc.
Query	A request for information from the Data Warehouse posed by the user or tool operated by the user.
Relational Database Management System	Warehouse for purposes of supporting Informational Applications.
Relational Database Management System (RDBMS)	A database system built around the relational model based on tables, columns and views.
Replication	The process of keeping a copy of data.
Source Database	The database from which data will be extracted or copied into the Data warehouse.

<b>Term</b>	<b>Definition</b>
Star Schema	A modeling scheme that has a single object in the middle connected to a number of objects around it radial manner - hence the name star. A fact (such as sales, compensation, payment, or invoices) is qualified by one or more dimensions (such as by month, by product, by geographical region). A fact table represents the fact and dimension tables represent the dimensions.
Target Database	The database in which data will be loaded or inserted.
Technical Directory	The portion of the Metadata Repository which deals with the technical information about the data. Such information may include the field designation (alphanumeric, etc.), the number of characters, range checks, etc. Warehouse for purposes of supporting Informational Applications.

## 15.2. Acronyms

Table 14 – List of Acronyms

Acronym	Description
ADM	Axiom Data Network
AEP	Advanced External Procedure
AIX	Advanced Interactive Executive
API	Application Programming Interface
ASP	Active Server Page
CD	Compact Disk
CDS	Central Data System
CFO	Chief Financial Office
CPU	Central Processing Unit
DBA	Database Administrator
DLL	Dynamic Link Library
DSN	Data Source Name
DWA	Data Warehouse Architecture
EDW	Enterprise Data Warehouse
ERP	Enterprise Resource Planning
ESQL	Embedded Structured Query Language
ETL	Extract, Transform and Load
EXL	Extensible Markup Language
FMS	Financial Management System
FTP	File Transfer Protocol
GB	Gigabyte
GUI	Graphical User Interface
HPUX	Hewlett-Packard UNIX
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IBM	International Business Machines
IE	Internet Explorer
IP	Internet Protocol

Acronym	Description
ISQL	Interactive Structured Query Language
ISS	Internet Information Server
MB	Megabyte
MDAC	Microsoft Data Access Components
Mhz	Megahertz
MQ	Message Queuing
MS	Microsoft
MX	Metadata Exchange
ODBC	Open Database Connectivity
OLAP	On-Line Analytical Processing
RAM	Random-access Memory
RDBMS	Relational Database Management System
ROLAP	Relational On-Line Analytical Processing
SAN	Storage Area Network
SDK	Software Development Kit
SFA	Student Financial Assistance
SID	Security Identifier
SQL	Structured Query Language
TCP	Transmission Control Protocol
TX	Transformation Expression
UNIX	Universal Interactive Executive
URL	Uniform Resource Locator
VDC	Virtual Data Center

# Appendix A

## Programming Choices Detail Information

The following appendix provides additional detail about the items summarized in the Programming Choices section.

To obtain this kind of extensibility, the Transformation Exchange (TX) dynamic invocation interference is built into PowerCenter. Using TX, an External Procedure transformation can be created and bound to an external procedure that has been developed. External procedures can be bound to two kinds of external procedures:

- COM external procedures (Available on Windows NT only)
- Informatica external procedures (Available on Windows NT and Solaris, Hewlett-Packard UNIX (HPUX), and Advanced Interactive Executive (AIX))

### **External Procedures and External Procedure Transformations**

There are two components to TX - External Procedures and External Procedure Transformations.

As its name implies, an external procedure exists separately from the Informatica Server. It consists of C, C++, or Visual Basic code, written by a user to define a transformation. This code is compiled and linked into a DLL or shared library, which is loaded by the Informatica server at runtime. An external procedure is bound to an External Procedure transformation.

An External Procedure Transformation is created in the Designer. It is an object that resides in the Informatica repository and serves several purposes:

- 1) It contains the metadata describing the external procedure. It is through this metadata that the Informatica server knows the signature (number and types of parameters, type of return value, if any) of the external procedure.
- 2) It allows an external procedure to be referenced in a mapping. By adding an instance of an external procedure transformation.
- 3) When a developer is creating Informatica external procedures, the external procedure transformation provides the information required to generate Informatica external procedure stubs.

### **Advanced External Procedures**

Advanced External Procedure (AEP) transformation create external transformation applications, such as sorting and aggregation, which require all input rows to be processed before emitting any output rows. To support these processes the input and output functions occur separately in the AEP. The AEP specified in the transformation is a separate callback function provided by Informatica that can be called from the AEP library. The output callback function is used to pass all the output port values from the AEP library to the Informatica server. In contrast, in the External Procedure transformation, an external procedure function does both input and output and its parameters consist of all the ports of the transformation.

## **PowerCenter.e:**

### *XML*

PowerCenter.e supplies two external command line programs, XML flattener (xmlflat) and XML Builder. The developer can use the programs to read and write XML data from within a PowerCenter session.

The XML Flattener enables the developer to:

- Extract metadata from an XML source document and create PowerCenter source definitions.
- Read XML data sources and use them as PowerCenter sources.

The XML Builder enables the developer to:

- Extract metadata from an XML target document and create flat file target definitions in PowerCenter.
- Construct an XML document from a flat file target.

### *MQ Series*

MQ Series is the messaging product from IBM that allows programs to communicate with one another across a network. The repository objects provided for MQ series and the executable program, MQ Listener, demonstrate how the developer can use PowerCenter.e to integrate with MQ Series. Using these components, the developer can:

- Get messages from an MQ Series message queue
- Put messages into an MQ Series message queue
- Monitor an MQ Series message queue and start a PowerCenter session when certain events occur.

Before transformation for the MQ Series can be used, the included sample code must be modified and recompiled. PowerCenter.e supplies the following Advanced External Procedure transformations as samples or templates that illustrate how the developer can communicate with the MQ series server from a PowerCenter session:

- AEP\_MQ\_GET
- AEP\_MQ\_PUT

These transformations are part of the sample repository included with PowerCenter.e. During installation, the libraries they reference are copied directly to the PowerCenter Server directory specified by the variable \$PMExtProcDir. The sample mappings for MQ Series use them to get and put messages into the Postcard queue. These procedures can be used as models for developing individual AEP's to communicate with MQ Series.

## **Reading Web Logs**

PowerCenter.e includes a set of pre-built mappings and mapplets that allow the developer to source web log data. Using these repository objects, the developer can deploy data models that extract that data and make it available for real-time analysis in a data warehouse. Since PowerCenter.e is built with the core PowerCenter platform, web data can be combined with any other data in the environment.

Web logs are complicated flat files that contain fairly fixed fields to track requests and web traffic. These log files are created on both Intranet and Internet web servers. The PowerCenter.e procedures for web logs parse the most common formats of web log files and enable the developer to use the web logs as data sources in PowerCenter mappings. PowerCenter.e supports these web log files, which contain fixed format records:

- Netscape web server
- Apache web server
- Microsoft Internet Information Server (IIS) web server, common and extended log formats.

## **Perl**

The Perl component of PowerCenter.e provides an Advanced External Procedures that a developer can use to call any Perl procedure from within a PowerCenter mapping. Using this AEP, a developer can easily incorporate the strong manipulation and program integration that might be needed to process various types of web log and web server data.

From a Perl procedure called within a PowerCenter.e mapping, the developer can invoke any number of Perl subroutines. For example, the developer might call a C routine from Perl to invoke the Informatica callback function directly from Perl. The developer could do this rather than returning values and invoking the callback in the C wrapper. Perl is probably an optimal solution for string manipulation.