Tivoli software

IBM

Tivoli Data Warehouse Version 1.3: Planning and Implementation

- A first look at Tivoli Data Warehouse Version 1.3
- Planning, implementing, and troubleshooting
- Performance optimization

Vasfi Gucer
Wolfgang Bergbauer
Marcel Berkhout
Thomas Bodenheimer
Andre Mello

Archived

Redbooks

ibm.com/redbooks

Archived

International Technical Support Organization

Tivoli Data Warehouse Version 1.3: Planning and Implementation

April 2005

Archived

SG24-6343-00

Note: Before using this information and the product it supports, read the information in "Notices" on page ix.

Archived

First Edition (April 2005)

This edition applies to IBM Tivoli Data Warehouse Version 1.3.

© Copyright International Business Machines Corporation 2005. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Archived

© Copyright IBM Corp. 2005. All rights reserved.

Archived

Archived

Archived

Archived

Archived

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

© Copyright IBM Corp. 2005. All rights reserved.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX 5L™	ibm.com®	Redbooks™
AIX®	IBM®	RUP®
Cloudscape™	Informix®	Tivoli Enterprise Console®
DB2 Universal Database™	iSeries™	Tivoli Enterprise™
DB2®	NetView®	Tivoli®
developerWorks®	OS/390®	WebSphere®
Domino®	Rational Unified Process®	xSeries®
DRDA®	Rational®	z/OS®
@server®	Redbooks (logo) ™	

The following terms are trademarks of other companies:

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Archived

Preface

With IBM Tivoli® Data Warehouse, you can analyze historical trends from various Tivoli and customer applications. The Tivoli Data Warehouse infrastructure enables a set of extract, transform, and load (ETL) utilities to extract and move data from Tivoli application data stores to a central repository. The open architecture of Tivoli Data Warehouse also enables data from non-Tivoli applications to be integrated into its central repository. Data from the central repository can be extracted into data marts that pertain to the reporting needs of selected groups. These data marts can also be used to produce cross-application reports.

This IBM Redbook focuses on planning, installation, customization, use, maintenance, and troubleshooting topics related to the new features of Tivoli Data Warehouse Version 1.3. This is done using a number of case study scenarios and warehouse packs.

This book has a comprehensive chapter about how to optimize the overall performance of a Tivoli Data Warehouse implementation, which focuses on DB2® optimization techniques and remote warehouse agents.

Also included with the book is a sizing spreadsheet that will help you determine the database sizing requirements of Tivoli Data Warehouse and various warehouse packs.

Finally, this book provides handy information and tips about troubleshooting Tivoli Data Warehouse.

We believe that this IBM Redbook will be a main reference for IT specialists who deploy Tivoli Data Warehouse at customer sites.

The team that wrote this Redbook

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

Vasfi Gucer is an IBM Certified Consultant IT Specialist working at the ITSO, Austin Center. He worked with IBM Turkey for 10 years and has been with the ITSO since January 1999. He has more than 10 years experience in systems management, networking hardware, and distributed platform software. He has

© Copyright IBM Corp. 2005. All rights reserved.

worked on various Tivoli customer projects as a Systems Architect in Turkey and the U.S. Vasfi is also a Certified Tivoli Consultant.

Wolfgang Bergbauer started his career as a Trainer (Certified Novell Instructor) in the areas TCP/IP, UNIX® and Linux®, Novell, and Internet security. He worked for several years for McKinsey&Company as a Network and System Management Specialist with a focus on enterprise-wide event management and correlation, and system management Web portal solutions. Currently, Wolfgang is a self-employed Network and System Management Specialist. He is working on IT Infrastructure Library (ITIL) projects and Extended Enterprise Management solutions.

Marcel Berkhout is an IT specialist working for IBM in The Netherlands. After graduating in 1998, he started at IBM working for various projects in IBM Global Services. In 2001, he moved to the Software Group as a DB2 specialist in the Software Group Services team. Within Software Group Services, in line with its mission, Marcel focuses on technical activities in support of software license deals and helps customers deploy IBM software by bringing in expert knowledge. Since Tivoli Data Warehouse was introduced, he has successfully implemented it at several large customers in The Netherlands. Performance tuning for Tivoli Data Warehouse has been his special focus over the last year, tuning several large environments in EMEA.

Thomas Bodenheimer is a Tivoli Software Engineer working for IBM in Research Triangle Park. He has worked extensively on performance testing of Tivoli Enterprise™ products, including Tivoli Data Warehouse.

Andre Mello is an IT Specialist at IBM Brazil. Andre joined IBM in 1996. He works in the capacity and performance group in the Strategic Outsourcing Organization. His areas of expertise are sizing, capacity planning, and performance tuning of databases for UNIX and Windows® platforms. Before this role, he worked as Database Administrator for many projects for IBM clients in Brazil.

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

Edson Manoel, Scott Vetter, Elizabeth Barnes
International Technical Support Organization, Austin Center

Catherine Cook, John Cook, Laura Farley, Warren Gill, Gareth Holl, Kevin Kingsbury, Mike Mallo, Ben Matteson, Page Hite
IBM U.S.

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks™ to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review Redbook form found at:

  ibm.com/redbooks

- Send your comments in an Internet note to:

  Redbook@us.ibm.com

- Mail your comments to:

  IBM® Corporation, International Technical Support Organization
  Dept. JN9B Building 905
  11501 Burnet Road
  Austin, Texas 78758-3493

Archived

Archived

1

Introducing Tivoli Data Warehouse V1.3

This chapter provides a brief introduction to the concepts, technologies, and products behind Tivoli Data Warehouse and the new features that can be found in Version 1.3. We cover the following topics:

- Data warehousing basics
- Tivoli Data Warehouse
- What is new in Tivoli Data Warehouse V1.3
- Tivoli Data Warehouse architecture
- Benefits of using Tivoli Data Warehouse

© Copyright IBM Corp. 2005. All rights reserved.

1.1 Data warehousing basics

Data warehousing is the process of managing a data warehouse and its components, called data marts. This management process includes all the ongoing support needs of the refresh cycle, database maintenance, and continual refinements to the underlying data model. In addition to that, data warehousing can be thought of as a tool to enable and support business intelligence.

The concept of data warehousing carries several other important terms mentioned in the previous paragraph. Such terms will be explained in the sections to follow. They are:

- Data warehouse
- Data mart
- Business intelligence
- Data mining

1.1.1 Data warehouse

A data warehouse is the cohesive data model that defines the central data repository for an organization. An important point is that we do not define a warehouse in terms of the number of databases. Instead, we consider it a complete, integrated data model of the enterprise, regardless of how or where the data is stored.

A data warehouse is a collection of databases where data is collected for the purpose of being analyzed. This collection of databases can be formed by one or more databases. The defining characteristic of a data warehouse is its purpose. Most data is collected to handle a company's ongoing business. This type of data can be called *operational data*. The systems used to collect operational data are referred to as online transaction processing (OLTP).

A data warehouse collects, organizes, and makes data available for the purpose of analysis in order to give management the ability to access and analyze information about its business. This type of data can be called *informational data*. The systems used to work with informational data are referred to as online analytical processing (OLAP).

Bill Inmon coined the term *data warehouse* in 1990. His definition is as follows:

> “A (data) warehouse is a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management's decision-making process.”

Archived

These are the main types of data:

- Subject-oriented: Data that gives information about a particular subject instead of about a company's ongoing operations
- Integrated: Data that is gathered into the data warehouse from a variety of sources and merged into a coherent whole
- Time-variant: All data in the data warehouse that is identified with a particular time period

1.1.2 Data mart

A data mart is a repository containing data specific to a particular business group in an enterprise. All data in a data mart derives from the data warehouse, and all data relates directly to the enterprise-wide data model. Often, data marts contain summarized or aggregated data that the user community can easily consume.

Another way to differentiate a data warehouse from a data mart is to look at the data's consumers and format. IT analysts and canned reporting utilities consume warehouse data, whose storage is usually coded and cryptic. The user community consumes data mart data, whose storage is usually in a more readable format. For example, to reduce the need for complex queries and assist business users who might be uncomfortable with the SQL language, data tables could contain the de-normalized code table values.

A data mart contains a subset of corporate data that is of value to a specific business unit, department, or set of users. This subset consists of historical, summarized, and possibly detailed data captured from transaction processing systems, or from an enterprise data warehouse. It is important to realize that a data mart is defined by the functional scope of its users, and not by the size of the data mart database. In parallel to increasing data mart usage, the underlying databases will rapidly increase in size.

1.1.3 Business intelligence

Business intelligence (BI) is not business as usual. It is about making better decisions more quickly and easily.

Businesses collect enormous amounts of data every day: information about orders, inventory, accounts payable, point-of-sale transactions, and, of course, customers. Businesses also acquire data, such as demographics and mailing lists, from outside sources. Unfortunately, based on a recent survey, over 93% of corporate data is not usable in the business decision-making process today. This applies also to systems management, where data tends to be of more technical nature.

Consolidating and organizing data for better business decisions can lead to a competitive advantage, and learning to uncover and leverage those advantages is what business intelligence is all about.

The amount of business data is increasing exponentially. In fact, it doubles every two to three years. More information means more competition. In the age of the information explosion, executives, managers, professionals, and workers all need to be able to make better decisions faster. Because now, more than ever, time is money.

Much more than a combination of data and technology, BI helps you to create knowledge from a world of information. Get the right data, discover its power, and share the value. BI *transforms information into knowledge*. Business intelligence is the application of putting the *right information* into the hands of the *right user* at the *right time* to support the decision-making process.

Business-driving forces

It can be noted that there are some business-driving forces behind business intelligence, one being the need to improve ease-of-use and reduce the resources required to implement and use new information technologies. Other driving forces behind business intelligence include:

- The need to increase revenues, reduce costs, and compete more effectively

  Gone are the days when end users could manage and plan business operations using monthly batch reports and IT organizations had months to implement new applications. Today, companies need to deploy informational applications rapidly and provide business users with easy and fast access to business information that reflects the rapidly changing business environment. Business intelligence systems are focused toward end-user information access and delivery and provide packaged business solutions in addition to supporting the sophisticated information technologies required for the processing of today's business information.

- The need to manage and model the complexity of today's business environment

  Corporate mergers and deregulation means that companies today are providing and supporting a wider range of products and services to a broader and more diverse audience than ever before. Understanding and managing such a complex business environment and maximizing business investment is becoming increasingly more difficult. Business intelligence systems provide more than just basic query and reporting mechanisms; they also offer sophisticated information analysis and information discovery tools that are designed to handle and process the complex business information associated with today's business environment.

- The need to reduce IT costs and leverage existing corporate business information

  The investment in IT systems today is usually a significant percentage of corporate expenses, and there is a need not only to reduce this overhead, but also to gain the maximum business benefits from the information managed by IT systems. New information technologies such as corporate intranets, thin-client computing, and subscription-driven information delivery help reduce the cost of deploying business intelligence systems to a wider user audience, especially information consumers such as executives and business managers. Business intelligence systems also broaden the scope of the information that can be processed to include not only operational and warehouse data, but also information managed by office systems and corporate Web servers.

1.1.4 Data mining

Data mining is the process of extracting *valid, useful, previously unknown,* and *comprehensible* information from data and using it to make business decisions.

The organizations of today are under tremendous pressure to compete in an environment of tight deadlines and reduced profits. Legacy and lengthy business processes that require data to be extracted and manipulated prior to use will no longer be acceptable. Instead, enterprises need rapid decision support based on the analysis and forecasting of predictive behavior. Data warehousing and data mining techniques provide this capability.

Data mining can be defined as the extraction of hidden predictive information from large databases and is a powerful technology with great potential to help companies focus on the most important information in their data warehouses. After a Tivoli Data Warehouse has been established, data mining tools can then be used to predict future trends and behaviors, enabling businesses to make proactive, knowledge-driven decisions.

Data mining tools can answer business questions that traditionally were too time-consuming to resolve. These tools hunt databases for hidden patterns, finding predictive information that experts might miss because it lies outside their expectations.

The art of data mining is not trivial, and it can be similar to “finding the needle in the haystack.” In this case, the needle is that single piece of intelligence your business needs, and the haystack is the large data warehouse you built up over a period of time within your business.

Most companies already collect and analyze massive quantities of data. Data mining techniques can be implemented rapidly on existing software and hardware platforms to enhance the value of existing information resources and can be integrated with new products and systems as they are brought online.

Given databases of sufficient size and quality, data mining technology can generate new business opportunities by providing these capabilities:

- Automated prediction of trends and behaviors: Data mining automates the process of finding predictive information in large databases. Questions that traditionally required extensive hands-on analysis can now be answered quickly and directly from the data. A typical example of a predictive problem is targeted server performance. Data mining uses data on past critical events to identify the servers most likely to cause future critical problems. Other predictive problems include forecasting server outage and other forms of performance degradation that is likely to occur, given certain events.
- Automated discovery of previously unknown patterns: Data mining tools sweep through databases and identify previously hidden patterns in one step. An example of pattern discovery is the analysis of IBM Tivoli Monitoring data to identify seemingly unrelated events that are often received together.

1.2 Tivoli Data Warehouse

Tivoli Data Warehouse V1.3 is built on IBM DB2 Data Warehouse. It offers all the IBM DB2 Data Warehouse functionality with additional Tivoli-specific extensions.

IBM Data Warehouse Management uses IBM DB2 Universal Database Enterprise Edition and the IBM DB2 Data Warehouse Manager feature. It provides an integrated, distributed, heterogeneous warehouse management infrastructure for designing, building, maintaining, governing, and accessing highly scalable, robust data warehouses, operational data stores, and data marts stored in IBM DB2 databases.

IBM DB2 Data Warehouse Manager helps warehouse administrators:

- To manage data volumes, to move data directly from source to target (also allowing packaged and simplified access to popular partner products such as SAP R/3), and to control the servers on which transformations take place with distributed warehouse agents
- To speed warehouse and data mart deployment with commonly used, pre-built data cleansing and statistical transformations
- To build and manage from a central point of control, integrated in IBM DB2, using the Data Warehouse Center graphical user interface

DB2 warehouse management consists of:

- An administrative client to define and manage data warehousing tasks and objects, and warehouse or data mart operations: the *Data Warehouse Center*
- A manager to manage and control the flow of data: the *warehouse server*
- Agents residing on IBM DB2 Universal Database Enterprise Edition server platforms to perform requests from the manager or warehouse server: the *local or remote warehouse agent*
- A *warehouse control database* storing the warehouse management metadata on a IBM DB2 database server
- A metadata administrative and publishing tool with its own administration graphical user interface (GUI): *Information Catalog Manager* to manage and present both technical and business metadata

The different components of the IBM DB2 Data Warehouse Manager are shown in Figure 1-1.

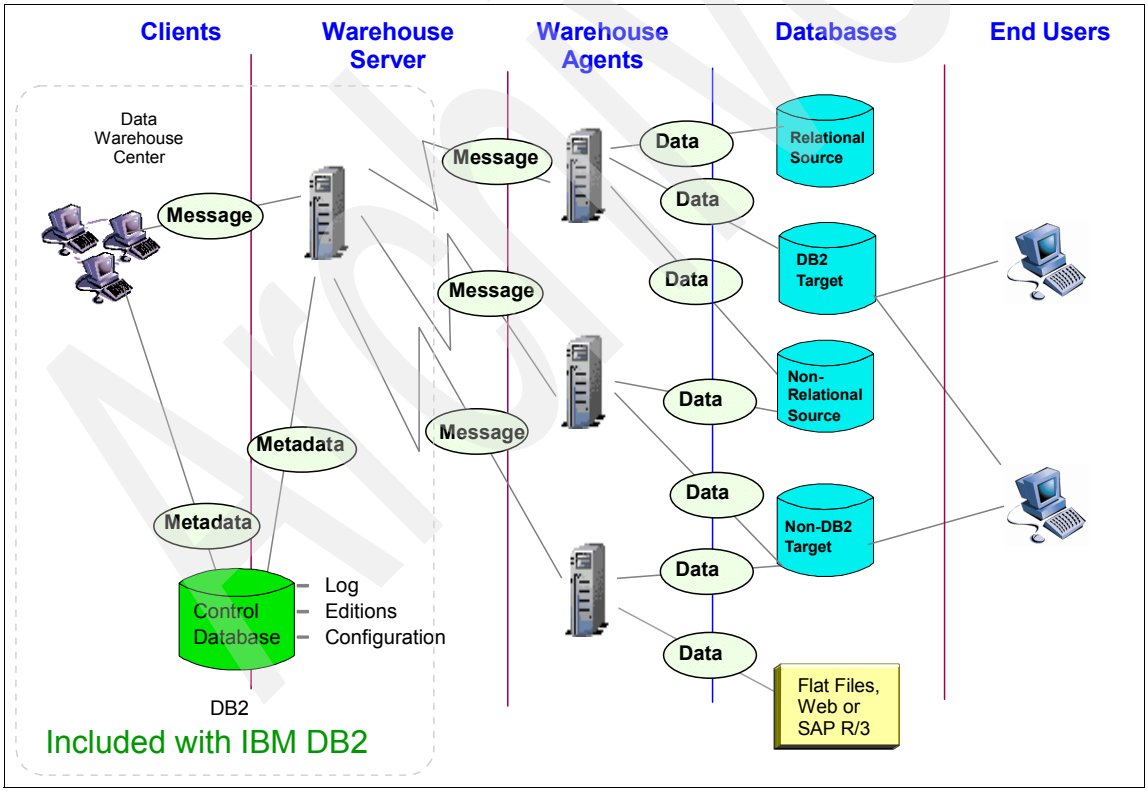


Figure 1-1 IBM DB2 Data Warehouse Manager

1.3 What is new in Tivoli Data Warehouse V1.3

Tivoli Data Warehouse V1.3 provides a number of enhancements and new features over Version 1.2, such as:

- Crystal Enterprise 10 on Microsoft® Windows, AIX®, Sun, and Linux: Crystal Enterprise 10 is now the supported release for Tivoli Data Warehouse V1.3. Tivoli Data Warehouse V1.3 is shipped with versions of Crystal Enterprise 10 for Windows, AIX, Solaris, and Linux.

  The Microsoft Windows version comes in two flavors, a full version that runs under a Web server (for example, Apache or IIS) and a version that runs under a servlet engine (for example, IBM WebSphere® or Tomcat). The UNIX (AIX, Solaris, Linux) versions only come in one flavor, a full version that runs under a servlet engine.

  Previous support for Crystal Enterprise 9 has been withdrawn and will need to be upgraded to Crystal Enterprise 10 before installing Tivoli Data Warehouse V1.3.

- Installing preconfigured central data warehouse and data mart databases: Central data warehouse and data mart databases can now be created outside of the Tivoli Data Warehouse V1.3 installation and then added to the Tivoli Data Warehouse V1.3 environment. This is done so that database administrators can set up the database to fit the performance needs of their specific environment. This is performed by running a series of configuration scripts to set up the central data warehouse or data mart database and schema. Then, the installation is run, and while specifying the central data warehouse or data mart database, you also need to specify that this database is a preconfigured database. The database is then added to the Tivoli Data Warehouse configuration without modifying the way that the database was setup.

- New platform support: See Table 1-1 on page 9 for platforms supported by Tivoli Data Warehouse V1.3.

- DB2 Version 8.2: Tivoli Data Warehouse Version 1.3 supports DB2 Version 8.2. This version also corresponds to DB2 Version 8.1 Fix Pack 7. DB2 Version 8.2 is shipped with Tivoli Data Warehouse V1.3.

Archived

Table 1-1 Platform support by Tivoli Data Warehouse component

Operating system	Data source	Warehouse agent	Control server	Central data warehouse	Data mart	Crystal publisher	Crystal Web server
Microsoft Windows 2000 Server SP2, Windows 2000 Advanced Server SP2, Windows Server 2003 Enterprise and Data Center Editions	Yes	Yes	Yes	Yes	Yes	Yes	Yes
IBM AIX 5L™, including Version 5.1, 5.2	Yes	Yes	No	Yes	Yes	Yes	Yes
Sun Solaris Solaris 8, Solaris 9, Solaris 10	Yes	Yes	No	Yes	Yes	No	2.8 only
Red Hat 7.2, 7.3	Yes	No	No	No	No	No	Yes
Red Hat Enterprise Linux 2.1 for xSeries®-32b, RHEL 3.0 AS+ for xSeries (ES and WS versions)	Yes	No	No	No	No	No	Yes
SUSE SLES 7.0, 7.2 7.3 xSeries, SLES 8.0/ UL 1.0 SP32 for xSeries SUSE	Yes	No	No	No	No	No	Yes
IBM z/OS® 1.2, 1.3, 1.4, 1.5	Yes	No	No	Yes	Yes	No	No

Archived

1.4 Tivoli Data Warehouse architecture

Tivoli Data Warehouse V1.3 is made available by Tivoli to consolidate historical data from different management applications into one or more central data warehouse databases and to make it easy to analyze *comprehensible*

information from data stored into its data mart databases, thus facilitating the business decision-making process. Figure 1-2 shows how the Tivoli Data Warehouse technology together with Crystal Enterprise can be used to provide an end-to-end business intelligence enabler.

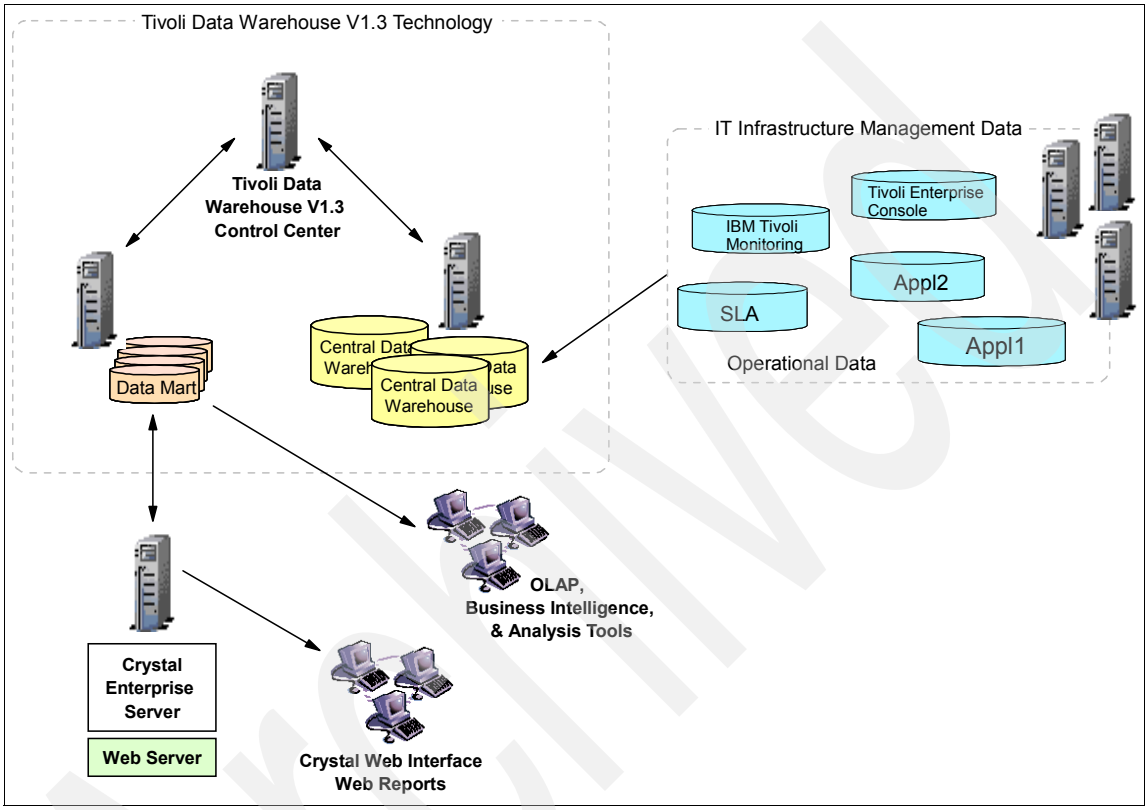


Figure 1-2 Tivoli Data Warehouse: The big picture

A Tivoli Data Warehouse V1.3 architecture can be composed of the following elements:

- Tivoli Data Warehouse control center server
- One or more central data warehouse databases
- One or more data mart databases
- IBM DB2 warehouse agents and agents sites
- Crystal Enterprise server

The following sections describe these elements in detail.

1.4.1 Tivoli Data Warehouse control center server

The control center server is the system that contains the control database for Tivoli Data Warehouse and is the system from which you manage your data. The control database contains metadata for both Tivoli Data Warehouse and for the warehouse management functions of IBM DB2 Universal Database™ Enterprise Edition. There can only be one control server in a Tivoli Data Warehouse V1.3 deployment.

1.4.2 Source databases

A source databases holds operational data to be loaded into the Tivoli Data Warehouse environment. Typically, the source databases are application specific and their number is likely to increase for a data warehouse installation.

Most Tivoli products provide a warehouse pack that makes application-specific data available in a source database. This can be a dedicated warehouse source database because it comes with IBM Tivoli Monitoring, or it can be an interface to the application's built-in database as provided for IBM Tivoli Storage Manager; IBM Tivoli NetView® warehouse packs for Tivoli products also include a means to upload data from the source database to the central data warehouse, thus minimizing the efforts for data collection.

1.4.3 Central data warehouse

The central data warehouse is a set of IBM DB2 databases that contains the historical data for your enterprise. You can have up to four central data warehouse databases in a Tivoli Data Warehouse V1.3 deployment.

1.4.4 Data marts

A separate set of IBM DB2 databases contains the data marts for your enterprise. Each data mart contains a subset of the historical data from the central data warehouse that satisfies the analysis and reporting needs of a specific department, team, customer, or application. You can have up to four data mart databases in a Tivoli Data Warehouse V1.3 deployment. Each data mart database can contain the data for multiple central data warehouse databases.

A warehouse pack for a Tivoli application provides all the necessary means to fill data marts with their specific data.

Archived

Note: Prior to Tivoli Data Warehouse V1.3, warehouse packs were called warehouse enablement packs. We use the new convention throughout this book.

1.4.5 Warehouse agents and agent sites

The warehouse agent is the component of IBM DB2 Warehouse Manager that manages the flow of data between data sources and targets that are on different computers. By default, the control center server uses a local warehouse agent to manage the data flow between operational data sources, central data warehouse databases, and data mart databases. You can optionally install the warehouse agent component of IBM DB2 Warehouse Manager on a computer other than the control center server.

Typically, you place an agent on the computer that is the target of a data transfer. That computer becomes a remote agent site, which the Data Warehouse Center uses to manage the transfer of Tivoli Data Warehouse data. This can speed up the data transfer, as well as reduce the workload on the control server.

1.4.6 Crystal Enterprise server

Crystal Enterprise Professional for Tivoli gives us a new mechanism for obtaining the reports provided by the warehouse enablement packs. The installation and configuration of a Crystal Enterprise environment is not mandatory, but if you intend to use it, do the installation before you begin installing Tivoli Data Warehouse V1.3.

Crystal Enterprise 10 is now the supported release for Tivoli Data Warehouse V 1.3. Tivoli Data Warehouse V1.3 is shipped with versions of Crystal Enterprise 10 for Microsoft Windows, AIX, Solaris, and Linux.

The Windows version comes in two flavors, a full version that runs under a Web server (for example, Apache or IIS) and a version that runs under a servlet engine (for example, WebSphere or Tomcat). The UNIX (AIX, Solaris, Linux) versions only come in one flavor, a full version that runs under a servlet engine.

Archived

Figure 1-3 shows an overview of the Tivoli Data Warehouse V1.3 architecture and supported software components.

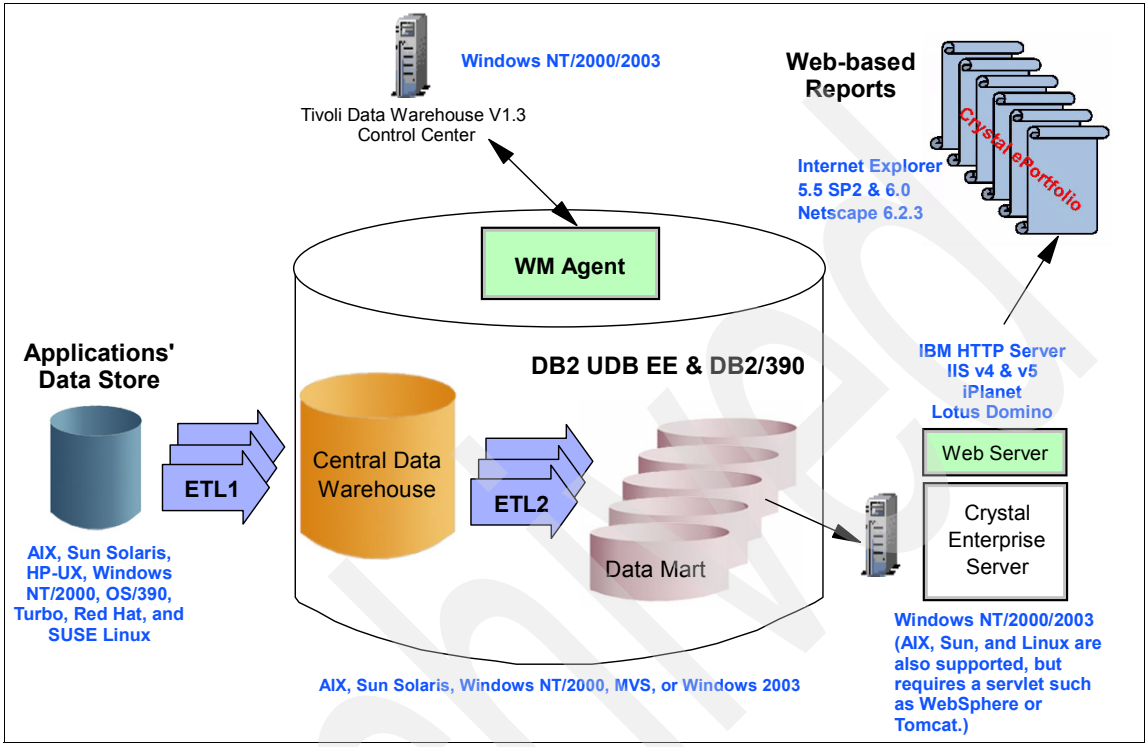


Figure 1-3 Detailed view of Tivoli Data Warehouse V1.3 components

1.5 Benefits of using Tivoli Data Warehouse

By using Tivoli Data Warehouse, you can access the underlying data about your IT infrastructure, including network devices and connections, desktops, hardware, software, events, and other information. With this information in a data warehouse, you can look at your IT costs, performance, and other trends across your enterprise. Tivoli Data Warehouse can be used to show the *value* and *return on investment* of Tivoli and IBM software, and it can be used to identify areas where you can be more effective.

With Tivoli Data Warehouse, you can see a consolidated view of the enterprise. This view can be used for:

- Server consolidation
- Capacity planning

- Under-used hardware discovery
- Disaster recover planning
- Service level advisor

As shown in Figure 1-4, Tivoli Data Warehouse V1.3 could be used as a single integration point for all systems management data, and it could also be used as both tool and technology to drive business intelligence within your enterprise.

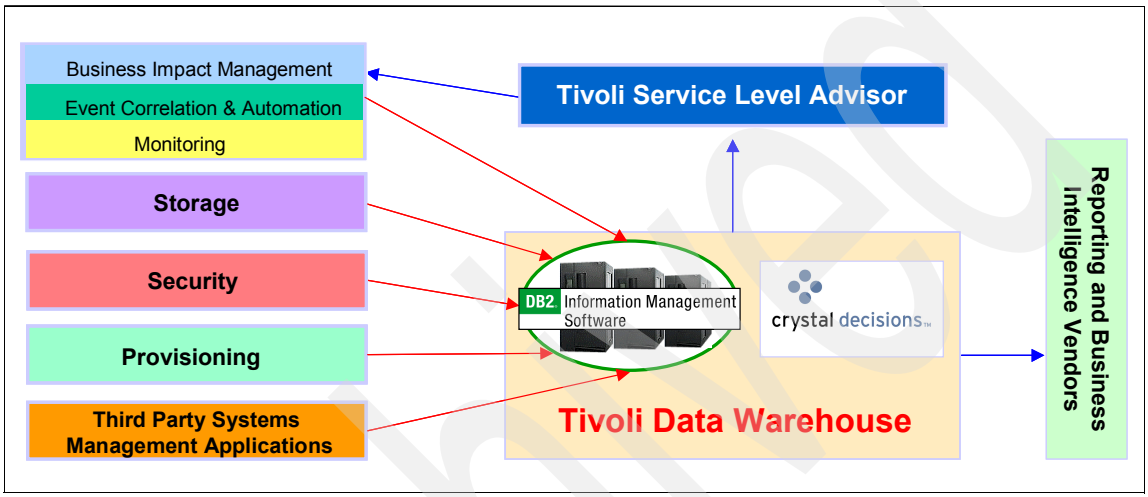


Figure 1-4 Tivoli Data Warehouse V1.3: A single integration point for all systems management data

Tivoli Data Warehouse provides the following capabilities:

- It has an open architecture for storing, aggregating, and correlating historical data.
- Most Tivoli products are shipped with warehouse packs to provide an easy and fast integration of the Tivoli products with Tivoli Data Warehouse.
- In addition to the data collected by diverse IBM Tivoli software, Tivoli Data Warehouse has the flexibility and extensibility to enable you to integrate your own application data.
- It offers database optimizations both for the efficient storage of large amounts of historical data and for fast access to data for analysis and report generation.
- It provides the infrastructure and tools necessary for maintaining the data:
  - These tools include the Tivoli Data Warehouse application, IBM DB2 Universal Database Enterprise Edition, Data Warehouse Center, DB2 Warehouse Manager, and Crystal Enterprise.

Archived

  - It includes the ability to use your choice of data analysis tools to examine your historical data.
  - In addition to the Crystal Enterprise reporting solution that is shipped with Tivoli Data Warehouse, you can analyze your data using any product that performs online analytical processing (OLAP), planning, trending, analysis, accounting, or data mining.
- It offers multicustomer and multicenter support:
  - You can keep data about multiple customers and multiple data centers in one warehouse, but restrict access so that customers can see and work with data and reports based only on their own data and not any other customer's data.
  - You can also restrict an individual user's ability to access data.
- It includes internationalization support:
  - Reports can be displayed in the language of the user's choice. Crystal Enterprise only comes in English, French, German, and Japanese; however, the reports generated by Crystal are translated into Brazilian Portuguese, French, German, Italian, Spanish, Japanese, Korean, Simplified Chinese, and Traditional Chinese.
  - This means that even if you are running the Crystal Enterprise server in English, you can view a report in Italian through the Crystal Report Viewer Web interface, if that language is set as your locale preference.

Archived

Archived

2

Planning a data warehouse project

This chapter analyses all the different tasks that lead to a successful data warehouse project. These tasks should be considered as a basic guideline of how to introduce, plan, scope, and architect a data warehouse project and, most importantly, how to make the project successful.

We look at the various stages of the project, which will mostly derived from well-proven concepts such as the Rational® Unified Process® (RUP®) and IT Infrastructure Library (ITIL).

It is very important to understand that we are discussing a data warehouse project and not only the Tivoli Data Warehouse component. The main focus of this chapter is to show the whole picture. The Tivoli Data Warehouse itself is a key part of this overall picture, but at the same time, it is just one part among several others that makes up a complete data warehouse solution.

The last part of this chapter shows a case study that uses the above principles to implement a data warehouse for a Internet Web Shop environment.

This chapter covers the following topics:

- Introduction
- The inception phase
- The elaboration phase

© Copyright IBM Corp. 2005. All rights reserved.

- The construction phase
- The transition phase
- The iteration
- The case study
- The data warehouse project
- Summary

Archived

2.1 Introduction

In almost all companies around the globe, the IT sector is always meant to support the key business processes or even make them available by giving them the platform on which to run. Therefore, the main reason for all system and network management implementations is to improve the quality of the IT environment by introducing well thought through and outlined processes. The individual disciplines within IT are outlined in several best practices, such as the IT infrastructure library, and should be clearly understood before starting a data warehouse project.

What is the main difference from a classical data warehouse solution?

The main difference lies not so much in the overall architecture, implementation, or delivery process of the data warehouse, but more in the sense of what systems or processes are supported by the data warehouse. The classic data warehouses mostly contain data that is directly related to the core business itself, for example, sold items during a certain time frame in the retail sector. Therefore, it can directly support key decisions for the core business of companies.

Data warehouses in the IT area most likely do not support the core business directly, but indirectly by improving the quality of the systems or processes on which the core business relies. If we use the retail sector as an example again, it could contain historical information about the transaction time a customer needs to order a specific item from a Web-based shopping system.

When we use the terms of the ITIL, the data warehouse mainly supports the following IT disciplines:

- Service-level management
- Capacity management
- Availability management
- Service continuity management

Important: The following outline of the data warehouse project phases is derived from the Rational Unified Process.

2.2 The inception phase

The following sections describe important considerations in the inception phase.

2.2.1 Understanding purpose and scope of data warehouse project

This statement sounds obvious, but it is really the key point of basically every project. All the project stakeholders must have a clear understanding of “why” a data warehouse is implemented. Reasons can be very simple and abstract, such as:

- Improve quality of IT support:

  The data warehouse can help the support organization within your IT organizations to do a better job by adapting support processes based on warehouse data.

- Enable mid-term to long-term IT capacity planning:

  The data within the warehouse gives real historical data about the utilization of your systems. This can help planning your IT environment for the future and reducing costs by eliminating unused capacity.

- Support IT managers:

  The data warehouse can provide decision makers with valuable data to support their strategic IT decisions. For example, just by the fact that you see the components and their performance of your business processes can help in rethinking and redesigning certain parts of your IT infrastructure.

- Deliver facts for decisions:

  This part should be considered very important, because many decisions are made on an instinctual level. With the real numbers of a data warehouse, this level can be raised. Real numbers will help in reasoning when it comes to IT budgets.

These are just a few examples that should help you in outlining the foundation of the data warehouse project. *Do not forget that if there are no real benefits, there is no reason to implement*.

2.2.2 Understanding the impact of the data warehouse

This second statement has to be seen in the context of the purpose of the data warehouse.

Only understanding the purpose does not mean that the IT organization is ready to take feedback from the data warehouse and incorporate this feedback in future decisions, plans, architectures, or organizational changes.

If the IT organization has no processes that are based on an iterative model, the data warehouse will be much less useful to the company. Figure 2-1 on page 21 outlines this iterative development, which originally was designed for software

development, but can also be used as a model for delivering complete business processes.

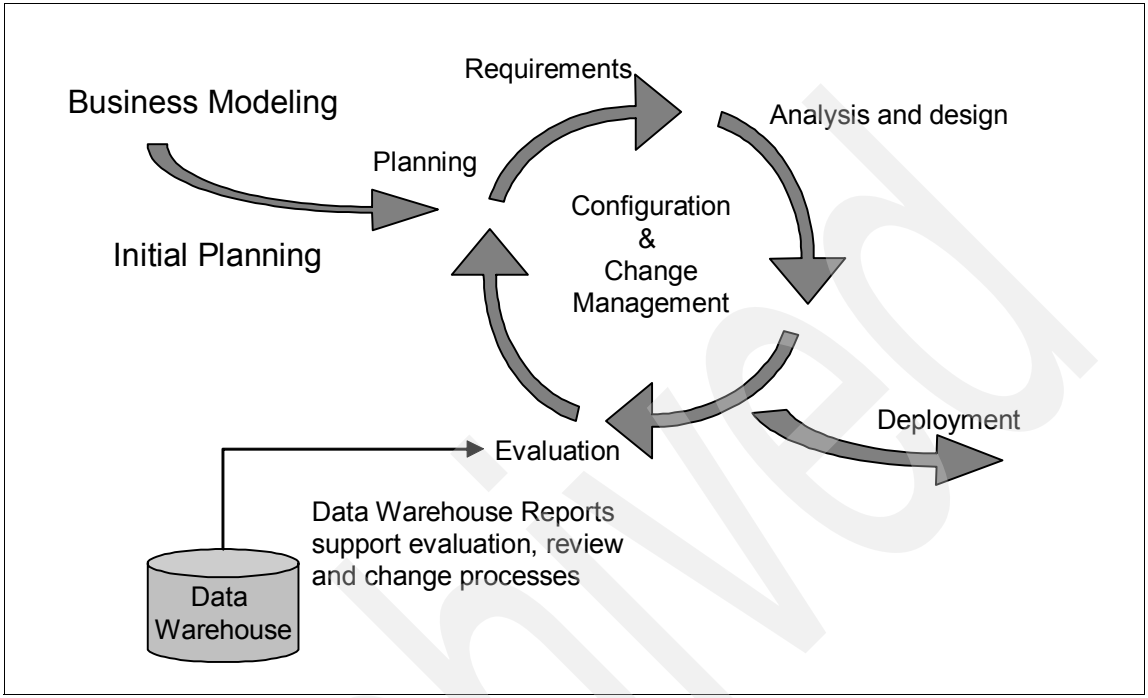


Figure 2-1 Iterative business process life cycle

Within this iterative business process life cycle, there is always the stage of review or reevaluation. This is the point where the data warehouse can play a key role in supporting decisions for any future changes to the business process.

2.2.3 Understanding the duration of a data warehouse project

Another important point is that the data warehouse effects and benefits are not immediate. This is a mid-term to long-term solution, and this must be understood by all stakeholders. Any effects seen within one year from building a business case can be seen as a major success, and not only require significant resources to implement the warehouse itself, but also require broad support within the IT organization.

Archived

After all stakeholders understand the key concepts and benefits of the data warehouse project and agree to a real implementation, the real data warehouse delivery process has to be outlined.

Do we need a prototype? When it comes to a new development, the question of a prototype is raised. We assume that a prototype is not necessary for a data warehouse project because a large part of the project is already covered by the Tivoli Data Warehouse product. It is useful to start with a very reduced set of sources for a small business case for a initial pilot project.

2.2.4 The business case

One of your most experienced IT analysts should be responsible for outlining a business case analysis.

The business case outlines one of your business processes and examines the various factors providing a data warehouse for this business process, for example:

- Identify unmet business needs for the case:

  For example, it could be that a business process is already monitored, but no capacity planing is possible due to the lack of historical information.
- Identify specific benefits for this case:

  That is, what specific business needs will be satisfied with the data warehouse. It is important to be specific.
- Identify all the costs required for implementing the data warehouse:

  It is important to have an estimate of the costs up front.
- Outline risks:

  Normally, the risks involved in a data warehouse are very small. There is little if any direct impact on the systems that implement the business process. The only serious risk is that the project itself fails.
- Identify what is needed for the data warehouse to have impact on the business process:

  This could be a monthly meeting with several IT groups or users to discuss the reports that are delivered by the reporting solution.

In addition, the feedback that has been collected during the discussion with the stakeholders can be used in outlining the business case.

Archived

2.3 The elaboration phase

The elaboration phase is broken down to the following sub-phases.

2.3.1 Outline the requirements

The various requirements have to be outlined for implementing the data warehouse. These requirements not only cover any logical components, but also physical components, resources needed, and all the participating IT groups or users. Examples are:

- Analyze the requirements of key users (for example, by interviews)
- Outline all users involved
- Design the logical structure of the data warehouse
- Outline all data sources and check for accessibility
- Estimate initial and mid-term sizing of the database
- Design the delivery process
- Outline the training requirements

2.3.2 Outline and present one technical blueprint

The final document should be a complete technical architecture document that outlines the whole data warehouse solution, including:

- Technical architecture
- Network and system infrastructure
- Physical design of databases and servers
- Hardware recommendations
- Logical design
- ETL processes
- Security concepts
- Backup and recovery processes

This document should not be seen as a perfect, final concept of the data warehouse. We strongly recommend that the iterative Rational Unified Process outlined for our first business case also be used for the whole data warehouse project and life cycle.

Archived

2.4 The construction phase

According to the technical blueprint, the individual tasks are performed by the various implementers of the data warehouse.

The result of this phase is a fully functional data warehouse solution.

2.5 The transition phase

Finally, the fully functional data warehouse solution is migrated to production. From now on, the various groups involved will use the data warehouse:

- Operations and support group:

  This group will be responsible of supporting all the different components that are needed to drive the data warehouse, for example operating system support, database support, and scheduling support.

- Development group:

  This group will support operations and support as a third-level contact. Any major issues that cannot be solved by operations and support have to be solved by development.

- Reporting group:

  The users who are preparing the reports generated from the data warehouse sources will get the first experience in using the delivered marts or similar sources such as OLAP cubes. (Of course, it is possible to see the reporting group as part of development as well, but reports and tools are very dynamic, so we see them as a separated, external item here.)

- Business process owners:

  Finally, the business process owners, who are eagerly getting the first reports that cover various components of their business process, will examine the first reports. They are responsible for about how the information derived from the report will influence their owned business process.

2.6 The iteration

After delivering the first production release, the iteration begins. We assume that there will be a few iteration phases before we can incorporate new business processes and expand the data warehouse from a small first stage environment to a real data warehouse.

Archived

The first iterations will deliver valuable data to improve the whole environment and to gain experience in regards to overall performance, capacity, and acceptance of your data warehouse environment.

2.7 The case study

The following sections outline a real live example. For a simple Web Shop application, we outline a data warehouse project. In these sections, we briefly describe the steps that need to be performed. Further details of the individual steps, such as installing Tivoli Data Warehouse and implementing a warehouse pack, can be found in the subsequent chapters of this book.

2.7.1 The Construction Limited company

Our virtual company is a very large retail firm in the construction and building sector that mainly buys and sells goods. We call it Construction Limited.

Within our firm, there is one department that is responsible for trading any kind of garden materials such as plants, tools, and chemicals for gardening.

This department has set up a Web-based shopping system, where customers can order any items from a catalog. It is a standard shopping cart application that is simple to use and similar to many Web-based shopping tools on the Internet. Figure 2-2 on page 26 shows the shopping cart application.

Archived

Figure 2-2 Standard shopping cart application

Because this shopping system is a very essential part of the key business of Construction Limited, the Web Shop is closely monitored to make sure that the application is up and running.

The internal IT department of Construction Limited is responsible for the development and for the management of the running applications. Several groups are involved managing the environment:

- Development department:

  This group develops the actual application with all the necessary components, but also the systems and components for monitoring the applications are developed within this group.

- Operations and support group:

  This group is responsible for deploying and running the applications.

- Change management group:

  This group tracks all the changes, does risk analysis, and closely works with the asset and configuration management group.

- Asset and configuration management group:

  This group is responsible for tracking all the hardware and software used, but they also maintain a configuration management database that is the main data source for many other IT applications and components.

- The reporting group:

  This group is a little outside of the IT group, but for several years, this group has been responsible for the central data warehouse that covers the core business of Construction Limited. Parts of this group will also be responsible for developing reports for the Web Shop applications.

A section of the organizational chart for the internal IT department of Construction Limited is shown in Figure 2-3.

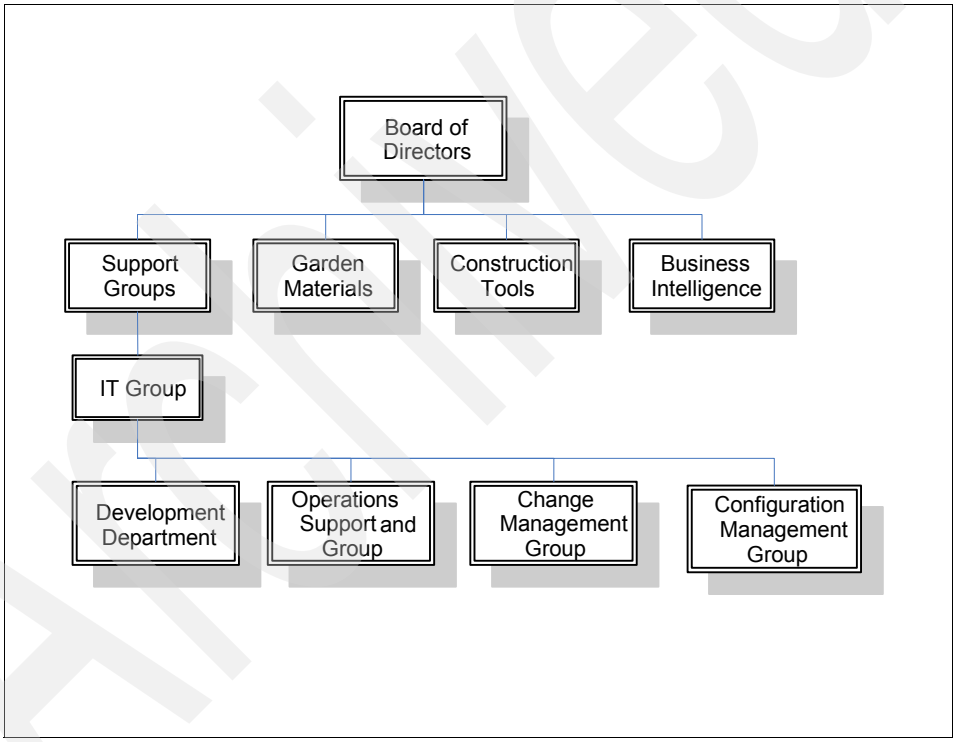


Figure 2-3 Section of the organizational chart

Archived

2.7.2 The Web Shop architecture

The Web Shop is based on several components that are shown in Figure 2-4:

- A DB2 database back end provides the storage of all the persistent data. It holds the order catalog, the actual orders, client information, and order status information.
- A WebSphere application server runs the actual application and provides Web-based interfaces to view items, order items, track the order status, and register new clients.
- Supporting components such as physical network, backup components, monitoring components, and failover software.

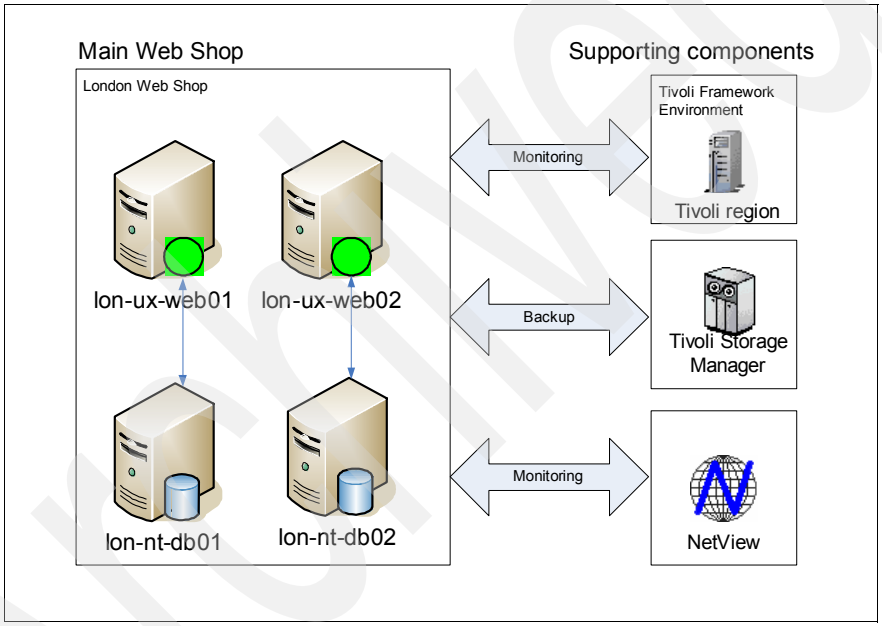


Figure 2-4 Web Shop components

2.7.3 Individual components

The Web Shop is composed by the hardware and software shown in the following tables.

Archived

Table 2-1 Web Shop hardware and software

Hardware				
Host name	**Description**	**Memory**	**CPU**	**Operating system**
milan	IBM @server pSeries 7025-F80	1GB	4	AIX 5L Version 5.2
helsinki	IBM @server pSeries 7043-150	512 MB	1	AIX 5L Version 5.2
tdw1301	IBM @server xSeries 7082-342	1 GB	2	Microsoft Windows 2000 Server
tdw1302	IBM @server xSeries 7082-342	1 GB	2	Microsoft Windows 2000 Server

Table 2-2 Software on the Microsoft Windows server

Software on Microsoft Windows 2000 Server	
Name	**Level**
IBM DB2 Universal Database Enterprise Edition V7.2	Fix Pack 11
Tivoli endpoint	Version 41100
IBM Tivoli Storage Manger Client	Version V5.1

Table 2-3 Software on the AIX server

Software on AIX 5L Version 5.2	
Name	**Level**
IBM DB2 Universal Database Enterprise Edition V7.2 Runtime Client	Fix Pack 11
IBM WebSphere Application Server, V5.0	Fix Pack 2
Tivoli endpoint	Version 41100
IBM Tivoli Storage Manager Client	Version V5.1

Archived

2.7.4 Monitoring environment

The Web Shop itself and all of its component are monitored from Tivoli Management Framework with several products installed:

- Tivoli Framework Version 4.1
- IBM Tivoli Monitoring Version 5.2
- IBM Tivoli Monitoring for Databases Version 5.1.1
- IBM Tivoli Monitoring for Web Infrastructure Version 5.1.2

Example 2-1 shows the full set of products and patches installed in Tivoli Management Framework.

Example 2-1 Products and patches installed in Tivoli Framework

```
*-----------------------------------------------------------------------------
*
                                Product List
*-----------------------------------------------------------------------------
*
Tivoli Management Framework 4.1.1
Tivoli ADE, Version 4.1.1  (build 11/20)
Tivoli AEF, Version 4.1.1  (build 11/20)
IBM Tivoli Monitoring for Databases, Version 5.1.0 - DB2
IBM Tivoli Monitoring Component Services, Version 5.1.1
IBM Tivoli Monitoring for Web Infrastructure - WebSphere Application Server
5.1.2
IBM Tivoli Monitoring - Tivoli Enterprise Data Warehouse Support, Version 5.1.2
Inventory Gateway, Version 4.2.1
Inventory, Version 4.2.1
Tivoli Java Client Framework 4.1
Tivoli Java Client Framework 4.1.1
Java 1.3 for Tivoli
Tivoli Java RDBMS Interface Module (JRIM) 4.1
Tivoli Java RDBMS Interface Module (JRIM) 4.1.1
JavaHelp 1.0 for Tivoli
IBM Tivoli Monitoring, Version 5.1.2
Distribution Status Console, Version 4.1.1
*-----------------------------------------------------------------------------
*
                                Patch List
*-----------------------------------------------------------------------------
*
Tivoli Framework Patch 4.1.1-TMF-0006  (build 02/18)
Tivoli Framework Patch 4.1.1-TMF-0007  (build 04/18)
Tivoli Inventory Server 4.2 Fix Pack 3, PTF U497604, 4.2-INV-FP03
Tivoli MDist 2 Graphical User Interface 4.1.1 001 Patch
IBM Tivoli Monitoring, Version 5.1.2 - Fix Pack 1
```

```
IBM Tivoli Monitoring Tivoli Enterprise Data Warehouse Support, Version 5.1.2 -
Fix Pack 1
IBM Tivoli Monitoring for Web Infrastructure 5.1.2 - WebSphere Application
Server - Fix Pack 02
IBM Tivoli Monitoring for Databases - DB2, Version 5.1.0 - Fix Pack 05
Scalable Collection Service, Version 4.1.1
```

Several default resource models have been distributed to the endpoints to monitor the following:

- Basic operating system data
- WebSphere health and availability
- DB2 health and availability

In addition, an IBM Tivoli NetView system is monitoring the availability of the devices. A SmartSet for the Web Shop devices has been created, as outlined in Figure 2.6.

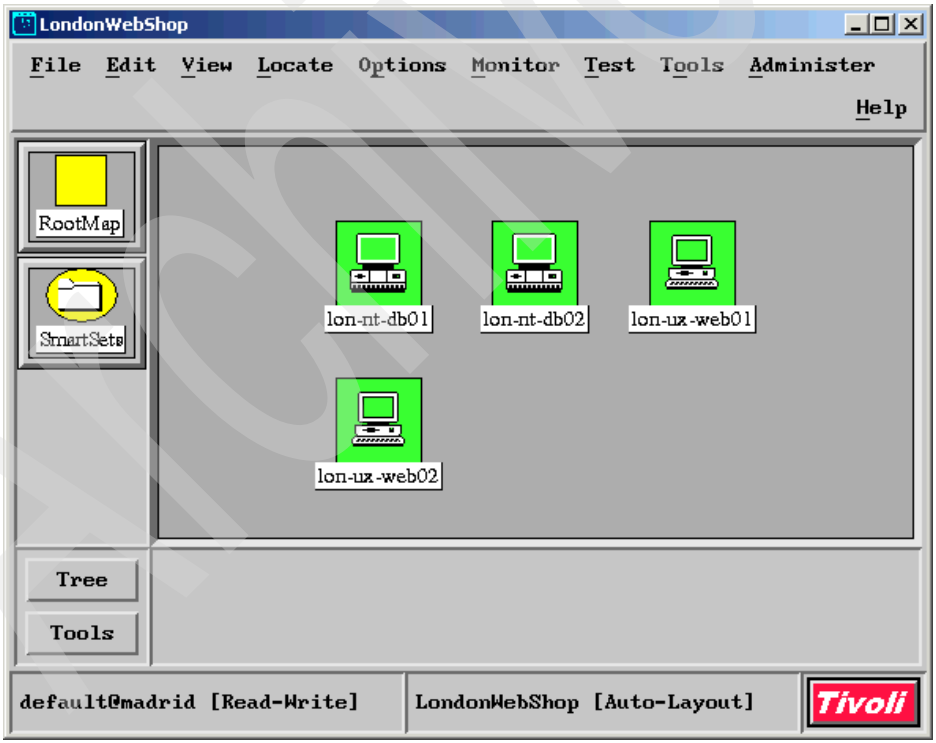


Figure 2-5 Monitoring the availability of the devices with NetView

Archived

For this case study, we assume that all the events from the Tivoli environment are forwarded to IBM Tivoli Enterprise Console®, so the operations and support group can create incidents within the help desk system from certain events.

2.8 The data warehouse project

The result of the following outlined iteration will be a complete functioning data warehouse system that is based on the default reports as outlined in other case study chapters. In a real world project, this first iteration could build the lab environment with the default reports.

After having this first iteration completed, the reporting team would be able to take the data and build custom reports designed to cover all the needs outlined in the requirements of this project.

2.8.1 The inception phase

Figure 2-6 shows the inception phase.

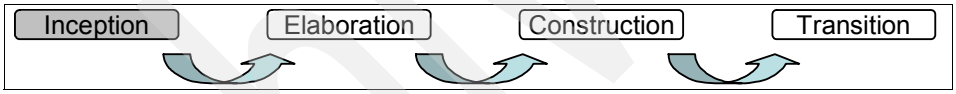


Figure 2-6 The inception phase

Because the Web Shop is a key component in the core business of Construction Limited, very often the question about customer satisfaction is raised. There are several ways of measuring the customer satisfaction, for example, by online surveys or telephone interviews. Because these methods reflect a very subjective side of the overall performance of the Web Shop, the managers of the Garden Materials group would like to have some objective performance data, which combined with the online surveys, could deliver input for key performance indicators (KPIs).

These KPIs would be summarized in reports and would be a main factor of influence about the future development and direction of the Web Shop. In addition, customer satisfaction could be tracked with a standard process and continuos improvements based on the reports could be made.

With the monitoring already in place, management decides to create a data warehouse solution that collects historical data and enables business intelligence to create the required KPI reports.

Construction Limited decides to phase a data warehouse project for the Garden Group Web Shop, having the future perspective in mind that this data warehouse could be expanded to support all major business processes that are based on the IT infrastructure of the company.

A board of stakeholders is defined and first meetings are scheduled.

After several meetings and discussions, all the members understand the key purpose and scope and impact of the data warehouse project, which can be summarized as:

- Measure the quality of the Web Shop:

  By measuring the transaction performance and availability of the Web Shop, the overall quality can be measured.
- Support decision makers:

  By outlining the KPI reports, decisions can be made for future developments.
- Deliver facts for decisions:

  Facts are much better arguments than numbers based on an instinctual level when it comes to budget planning and new requirements for a business process.
- Enable mid-term to long-term IT capacity planning:

  Having all the historical information about performance and utilization, much better assumptions about future needs in regards to hardware and software spending can be made.

2.8.2 The elaboration phase

Figure 2-7 shows the elaboration phase. Because all stakeholders are in agreement and fully understand the purpose and scope of the project, and also the funding has been approved, we are ready to go to the drawing board.

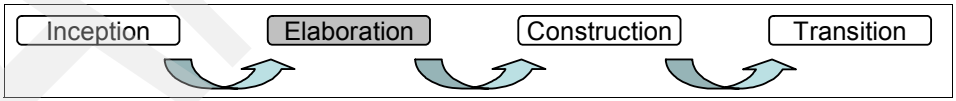


Figure 2-7 The elaboration phase

Architecture

We use a very simple architecture for this first data warehouse project, shown in Figure 2-8 on page 34. The idea is to have one data warehouse database server, including the control server, one server for the Crystal Enterprise software, and one additional database server holding the source databases.

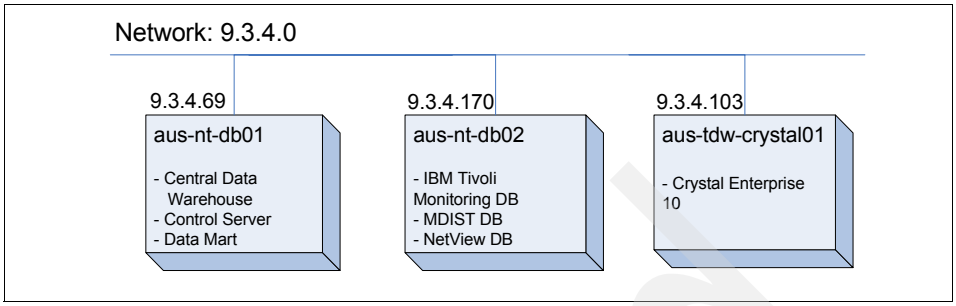


Figure 2-8 Network components

Hardware requirements

Table 2-4 shows the hardware requirements.

Table 2-4 Hardware requirements

Host name	Platform	Recommended size
aus-nt-db01	Windows/x86	2.0 GB RAM 2.4 GHz processor 36 GB disk RAID 1 5x 36GB RAID 5
aus-nt-db02	Windows/x86	1.0 GB RAM 2.4 GHz processor 2x 36 GB System RAID 1 5x 36 GB RAID 5
aus-tdw-crystal01	Windows/x86	1.0 GB RAM 2.4 GHz processor 2x 36 GB System RAID 1 2x 36 GB Appl. Partition RAID 1

Software requirements

The software that is installed on the systems is provided in the following tables.

Table 2-5 on page 35 shows the software components on aus-nt-db01.

Archived

Table 2-5 Software components on aus-nt-db01

Component	Software	Versions
Operating system	Windows 2000 Server	Serivce Pack 4
Database	DB2	DB2 Server 7.2 FP11
Tivoli Data Warehouse product	Tivoli Data Warehouse	Version 1.2 + Fix Pack 02
Warehouse Enablement Packs aus-nt-db01	IBM Tivoli NetView Warehouse Enablement Pack	Version 1.1.0 + Fix Pack 1
	IBM Tivoli Monitoring, Version 5.1.1, Warehouse Enablement Pack	Version 5.1.1.600 + Version 5.1.1.670
	IBM Tivoli Monitoring for Operating Systems, Version 5.1.1, Warehouse Enablement Pack	Version 1.1.0.600 + Version 5.1.1.670
	IBM Tivoli Monitoring for Databases, Version 5.1.2: DB2, Warehouse Enablement Pack	Version 1.1.0 + Version 1.1.0.1
	IBM Tivoli Monitoring for Web Infrastructure, Version 5.1.0: WebSphere Application Server, Warehouse Enablement Pack	Version 1.1.0 + Version 1.1.0.3

Table 2-6 shows the software components on aus-nt-db02. This server holds only a DB2 database.

Table 2-6 Software components on aus-nt-db02

Component	Software	Versions
Operating system	Windows 2000 Server	Service Pack 4
Database	DB2	DB2 Server 7.2 FP11

Table 2-7 on page 36 shows the software components on aus-tdw-crystal01. This server holds only a DB2 database.

Archived

Table 2-7 Software components on aus-tdw-crystal01

Component	Software	Versions
Operating system	Windows 2000 Server	Service Pack 4
Reporting	Crystal Enterprise	Version 9

2.8.3 The construction phase

During this phase, the development team will set up the systems according to the outlined architecture. All the software and components will be installed, configured, and tested in the lab environment. Figure 2-9 shows the construction phase.

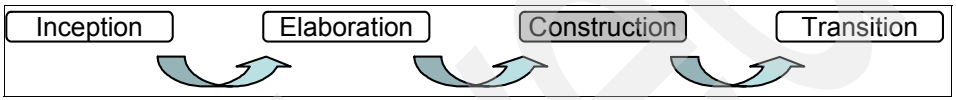


Figure 2-9 The construction phase

During the construction phase, solid documentation is essential for any further iterations in the whole life cycle. The project manager should make sure that proper documentation is delivered.

2.8.4 The transition phase

Figure 2-10 shows the transition phase.

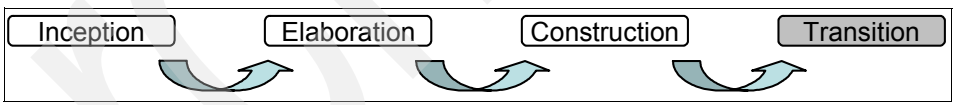


Figure 2-10 The transition phase

Within our first life cycle, the transition phase will move the data warehouse environment to a “semi-productional” state. That means that we will collect data from the production IBM Tivoli Monitoring databases, but the data warehouse and reporting environment are still seen as a lab environment. With this outline, we are able to use productional data to:

- Transfer operational knowledge to the support group
- Perform in-depth performance analysis
- Perform backup and restore tests
- Enable the reporting team to develop custom reports
- Develop a monitoring environment for the data warehouse environment

2.9 Summary

After finishing a complete RUP cycle and collecting productional data for several months, the next RUP cycle would move our example system to the real production world.

We think that for a successful data warehouse environment it is absolutely essential that you phase your project and do not underestimate the effort of maintaining a data warehouse for your operational IT data. It is essential that all participating members understand that the Tivoli Data Warehouse product itself is not a out-of-the-box installation that is shipped with the core Tivoli products. It is a complex environment that needs thorough understanding and commitment; otherwise, disappointment is inevitable.

Archived

Archived

3

Tivoli Data Warehouse design considerations

This chapter provides planning considerations for Tivoli Data Warehouse. We cover the following topics:

- Hardware and software requirements
- Physical and logical design considerations
- Database sizing
- Security
- Network traffic considerations
- Integration with other business intelligence tools
- ETL development
- Skills required for a Tivoli Data Warehouse project

© Copyright IBM Corp. 2005. All rights reserved.

3.1 Hardware and software requirements

Tivoli Data Warehouse V1.3 can be deployed in two different configurations:

- *Quick deployment installation*, also known as a quick start or stand-alone installation, with all the components installed on a single Microsoft Windows NT®, Windows 2000, or Windows 2003 system. This is convenient for demonstrations, as an educational or test platform, and for companies that do not plan to have many users concurrently accessing the data stored in the Tivoli Data Warehouse databases, those that do not need to capture and analyze large amounts of data, or both.
- *Distributed installation*, with the components installed on multiple systems in your enterprise, including UNIX and z/OS servers. See 3.1.2, "Software requirements" on page 42 to determine the operating systems supporting each component of Tivoli Data Warehouse.

The historical reporting for Tivoli Data Warehouse V1.3 is provided by Crystal Enterprise. Tivoli Data Warehouse V1.3 is shipped with versions of Crystal Enterprise 10 for Microsoft Windows, AIX, Sun Solaris, and Linux. This package is called *Crystal Enterprise Professional for Tivoli*. The Windows version comes in two flavors, a full version that runs under a Web server (for example, Apache or IIS) and a version that runs under a servlet engine (for example, WebSphere or Tomcat). The UNIX (AIX, Solaris, Linux) versions only come in one flavor, a full version that runs under a servlet engine.

There is also an expanded installation that enables you to install Crystal Enterprise server components on more machines in order to create an Automated Process Scheduler (APS) cluster, to increase available resources, and to distribute the processing workload. In order to achieve this configuration, you need to separately license Crystal Enterprise from Crystal Reports.

In the following sections, we provide the current hardware and software requirements for a Tivoli Data Warehouse environment, but you should also check the *Tivoli Data Warehouse Release Notes*, SC32-1399 for possible updates about these requirements.

3.1.1 Hardware requirements

This section provides information about the hardware requirements for installing Tivoli Data Warehouse V1.3 components. Table 3-1 on page 41 lists the recommended hardware requirements for Tivoli Data Warehouse V1.3 for both the stand-alone and distributed configurations.

Note: The values in Table 3-1 differ from the information provided in *Tivoli Data Warehouse Release Notes*, SC32-1399. They represent the hardware used at the time of writing this book and serve as our recommended starting configuration for both the stand-alone and distributed configurations.

Table 3-1 Hardware recommendations for Tivoli Data Warehouse components

Installation configuration	Tivoli Data Warehouse components	Recommended size
Stand-alone installation	All	1.0 GB RAM 2.4 GHz processor 30 GB disk
Distributed installation	Control server	512 MB RAM 1.3 MHz processor 10 GB disk
	Central data warehouse	512 MB RAM 1.3 MHz processor (400 MHz CPU or equivalent for UNIX) 30 GB disk
	Data mart	512 MB RAM 1.3 MHz processor (400 MHz CPU or equivalent for UNIX) 30 GB disk

Be aware that as the warehouse packs are added to the Tivoli Data Warehouse V1.3 installation, additional hard disk space is required. See the documentation provided by the warehouse pack for application planning information and additional hard disk space requirements.

See Chapter 4, “Sizing Tivoli Data Warehouse V1.3” on page 81 to help you in evaluating the storage required by the different metrics you want to collect in your own environment.

Archived

Table 3-2 lists the hardware requirements for additional components of a Tivoli Data Warehouse V1.3 solution.

Table 3-2 Additional hard disk space requirements

Component	Recommended hardware
Crystal Enterprise Professional Version 9 for Tivoli	1.0 GB RAM 2.4 GHz processor 30 GB disk (1 GB for installation alone)
Warehouse agent	50 MB disk

The storage required by the installation of Crystal Enterprise Professional Version 9 for Tivoli shown in Table 3-2 assumes a full stand-alone installation of Crystal Enterprise Professional Version 9 for Tivoli. Note that on the Crystal Enterprise server, additional storage is required for Web server, database client software, and all the reports installed by each warehouse pack.

3.1.2 Software requirements

Table 3-3 provides information about the software requirements for the Tivoli Data Warehouse V1.3 components.

Note: You might receive confusing error messages if your systems do not meet the software requirements listed in this section.

Table 3-3 Software requirements

Operating system	Data source	Warehouse agent	Control server	Central data warehouse	Data mart	Crystal publisher	Crystal Web server
Windows 2000 Server SP2, Windows 2000 Advanced Server SP2, Windows Server 2003 Enterprise and Data Center Editions	Yes	Yes	Yes	Yes	Yes	Yes	Yes
IBM AIX 5L, including Versions 5.1, 5.2	Yes	Yes	No	Yes	Yes	Yes	Yes

Operating system	Data source	Warehouse agent	Control server	Central data warehouse	Data mart	Crystal publisher	Crystal Web server
Windows 2000 Server SP2, Windows 2000 Advanced Server SP2, Windows Server 2003 Enterprise and Data Center Editions	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Sun Solaris Solaris 8, Solaris 9, Solaris 10	Yes	Yes	No	Yes	Yes	No	2.8 only
Red Hat 7.2, 7.3	Yes	No	No	No	No	No	Yes
Red Hat Enterprise Linux 2.1 for xSeries - 32b, RHEL 3.0 AS+ for xSeries (ES and WS versions)	Yes	No	No	No	No	No	Yes
SUSE SLES 7.0, 7.2 7.3 xSeries, SLES 8.0/ UL 1.0 SP32 for xSeries SUSE	Yes	No	No	No	No	No	Yes
z/OS 1.2, 1.3, 1.4, 1.5	Yes	No	No	Yes	Yes	No	No

3.1.3 Database requirements

Tivoli Data Warehouse V1.3 requires the IBM DB2 Universal Database Enterprise Edition V7.2. It is shipped with Tivoli Data Warehouse V1.3.

For warehouse databases on z/OS systems, the supported database version is DB2 Universal Database for OS/390 and z/OS V7.1 with the DB2 Management Clients Package installed (FMID JDB771D).

The recommended hard disk space in Table 3-1 on page 41 might not be large enough to accommodate some data growth as transactions are added to the

Archived

database. However, when you plan for your database requirements, you must consider the following possibilities:

- Future data growth
- Addition of warehouse packs

We recommend that you install your central data warehouse on an expandable system with a minimum of 30 GB of data space.

The data source support is determined by Open Database Connectivity (ODBC) support in IBM DB2 releases and specified by the warehouse pack product.

> **Note:** The actual RDBMSs supported as data sources for a given warehouse pack are documented in the *readme* file for that warehouse pack.

3.1.4 Crystal Enterprise requirements

The server components of Crystal Enterprise Professional Version 9 for Tivoli can be installed on the following operating systems:

- Microsoft Windows NT4 Server SP6a
- Microsoft Windows 2000 SP3, Server version
- Microsoft Windows 2000 SP3, Advanced Server version
- Microsoft Windows 2000 SP3, Data Center version
- Microsoft Windows 2003
- IBM AIX 5L Version 5.1 or 5.2 (see the following note)
- Sun Solaris 2.8 (see the following note)
- Red Hat 7.1, 7.2, Enterprise Linux 2.1 (see the following note)
- SUSE SLES 7.0, 7.2 7.3 xSeries, SLES 8.0/ UL (see the following note)

> **Note:** The UNIX (AIX, Solaris, Linux) versions run under a servlet engine such as WebSphere.

The Crystal Enterprise Automated Process Scheduler (APS) component requires a database to store information about the system and its users. By default, the Setup program installs and configures its own Microsoft Data Engine (MSDE) database if necessary. The MSDE is a client/server data engine that provides local data storage and is compatible with Microsoft SQL Server. Crystal Enterprise Professional for Tivoli APS clustering is automatically supported by the default MSDE database.

If you already have the Microsoft Data Engine or Microsoft SQL Server installed, the installation program creates the APS database using your existing database engine.

You can migrate this initial APS database to another database server later. Supported servers for APS database migration are:

- Microsoft SQL Server 7 SP4 (ODBC)
- Microsoft SQL Server 2000 SP2 (ODBC)
- Microsoft MSDE (ODBC)
- Oracle*9i* (native)
- Oracle*8i* (8.1.7) (native)
- IBM DB2 UDB 8.2 (native)
- Sybase Adaptive Server 12.5 (ODBC and native)
- Informix® Dynamic Server 2000 v 9.21 (ODBC)

Note: If the Microsoft Data Engine (MSDE) or Microsoft SQL Server is already installed on the local machine, you must set up a user account for the Crystal Enterprise Professional for Tivoli APS before installing Crystal Enterprise Professional for Tivoli, as follows:

1. Determine whether the Crystal Enterprise Professional for Tivoli APS should use Windows NT or Microsoft SQL Server authentication when connecting to your local database installation.
2. Using your usual administrative tools, create or select a user account that provides Crystal Enterprise with the appropriate privileges to your database server:
   - If you want the APS to connect to its database using Windows NT authentication, ensure that the Windows NT user account that you assign to the APS has System Administrator's role in your SQL Server installation.

     In this scenario, the Windows NT user account that you assign to the APS is not actually used to create the system database during the installation process. Instead, your own Windows NT administrative account is used to create the database, so verify that your Windows NT account also has the System Administrator's role in your SQL Server installation.
   - If you want the APS to connect to its database using SQL Server authentication, the logon information that you assign to the APS must belong to the Database Creators role in your SQL Server installation. In this scenario, the SQL Server credentials that you assign to the APS are also used to create the database and its tables.
3. Verify that you can log on to SQL Server and carry out administrative tasks using the account you set up for use by the APS.

For details about APS database migration, see “Configuring the intelligence tier” in the *Crystal Enterprise 10 Administrator’s Guide* manual, which is shipped with the product.

Note: For a detailed list of environments tested with Crystal Enterprise, consult the Platforms.txt file included with your product distribution.

In case you acquired a Crystal Enterprise 10 Special Edition license and choose to deploy the server-side installation method using the Crystal server connected to an external Web server, you also need to install and configure the appropriate Web Connector on your Web server machine. The supported Web servers are listed in Table 3-4.

Table 3-4 Web servers and OS supported by Crystal Web Connector

Web server	Operating system
Microsoft IIS5/ISAPIY Microsoft IIS5/CGIY	Microsoft Windows 2000 Server Microsoft Windows 2003 Server
Microsoft IIS4/ISAPI-Y Microsoft IIS4/CGI-Y	Windows NT 4.0 Server
iPlanet 6.0 SP3/NSAPI iPlanet 6.0 SP3/CGI iPlanet 4.1 SP10/NSAPI iPlanet 4.1 SP10/CGI	Windows 2000 Server Windows 2003 Server Windows NT 4.0 Server Sun Solaris 2.7, 2.8
Domino® 5.0.8/DSAPI	Windows 2000 Server Windows 2003 Server Windows NT 4.0 Server
Domino 5.0.8/CGI	Windows 2000 Server Windows 2003 Server Windows NT 4.0 Server Sun Solaris 2.7, 2.8 IBM AIX V4.3.3, AIX 5L Versions 5.1, 5.2
Apache 1.3.26/ASAPI Apache 1.3.26/CGI	Sun Solaris 2.7, 2.8 Red Hat 6.2/7.3 (x86) SUSE 7.3/8.0 (x86) IBM AIX V4.3.3, AIX Versions 5.1, 5.2
IBM HTTP 1.3.19.2/ASAPI	IBM AIX 4.3.3, 5.1, 5.2
IBM HTTP 1.3.19.2/CGI	Windows 2000 Server Windows 2003 Server IBM AIX V4.3.3, AIX 5L Versions 5.1, 5.2

Archived

Web browser requirements

Crystal Enterprise Professional for Tivoli supports the following Web browsers:

- Microsoft Internet Explorer 6.0 (Windows)
- Microsoft Internet Explorer 5.5 SP2 (Windows)
- Netscape 6.2.3 (Windows)
- Microsoft Internet Explorer 5.0 on OS9 or OS X Classic mode (Macintosh)

Note: Crystal Enterprise delivers reports and analytic content to any supported browser using pure DHTML.

3.2 Physical and logical design considerations

In the previous section, we briefly introduced the basic components of the Tivoli Data Warehouse V1.3 application. Now, we discuss how those components can be installed on different servers and the advantages or the possible drawbacks of the different configurations.

All the possible architectures for a Tivoli Data Warehouse implementation are determined by the location of the following components:

- Control server
- Central data warehouse
- Data marts
- Crystal Enterprise server
- Warehouse agents
- Source databases

All these components can coexist on the same server, or they can be spread out on different servers. See Table 3-5 to determine which operating systems are the platforms required by each Tivoli Data Warehouse component.

Table 3-5 Requirements for Tivoli Data Warehouse components

Component	Supported operating systems platform[a]	DB2 components required
Control server	Microsoft Windows	IBM DB2 Universal Database Enterprise Edition
Central data warehouse databases	Windows, AIX, Solaris, z/OS	IBM DB2 Universal Database Enterprise Edition IBM DB2 Warehouse Manager (optional)

Archived

Component	Supported operating systems platform[a]	DB2 components required
Data mart databases	Windows, AIX, Solaris, z/OS	IBM DB2 Universal Database Enterprise Edition IBM DB2 Warehouse Manager (optional)
Crystal Enterprise server	Windows, AIX	IBM DB2 Universal Database Enterprise Edition or IBM DB2 database client
Warehouse agents	Windows, AIX, Solaris	IBM DB2 Universal Database Enterprise Edition IBM DB2 Warehouse Manager

a. See the *Tivoli Data Warehouse Release Notes*, SC32-1399, for complete details about supported operating systems.

3.2.1 Source databases

The source databases are the fundaments of a data warehouse implementation. Their health is of crucial impact to the whole Tivoli Data Warehouse environment.

From an administrative point of view, source databases can be divided into two types:

- The first type of source database is part of the product from which data should be collected. An example is IBM Tivoli Storage Manager, where the source database is the internal Tivoli Storage Manager database. In this case, the application administrator should be responsible for this source database.
- The second type, such as the IBM Tivoli Monitoring source database, is built for the purpose of a source database exclusively. In this sense, you can consider these as part of the data warehouse infrastructure.

3.2.2 Control server

The control server is the system that contains the control database for Tivoli Data Warehouse and from which you manage your data warehouse environment. Only one control server can be installed in a Tivoli Data Warehouse environment and it must be installed on a Windows platform.

Before installing the control server, you must install IBM DB2 Universal Database Enterprise Edition V7.2 locally and have the Crystal Enterprise environment up and running on the same or on a different server. You must also install IBM DB2 Universal Database Enterprise Edition V7.2 or DB2 Universal Database for OS/390 and z/OS V7 on each computer that will contain a central data warehouse database or a data mart database.

The control server creates a *control database* (TWH_MD) that contains descriptions of the stored data (known as *metadata*) for both Tivoli Data Warehouse and for the warehouse management functions. The control server uses the DB2 Data Warehouse Center to automate the data warehouse processing and to define the ETL processes that move and transform data into the central data warehouse and the data marts.

The control server also runs a warehouse agent, the component of IBM DB2 Warehouse Manager that manages the data flow between warehouse sources and targets. In advanced Tivoli Data Warehouse scenarios, you can move the warehouse agent to other locations (see 3.2.8, “Warehouse agents” on page 61).

When using the configuration with the warehouse agent on the control server, the computer on which you install the control server must also connect to the operational data stores of your enterprise, which potentially reside on other systems and in relational databases other than IBM DB2. To enable the control server to access these data sources, you must install the appropriate database client for each data source on the control server system.

3.2.3 Central data warehouse

The central data warehouse is a set of IBM DB2 databases containing the historical data for your enterprise.

The central data warehouse does not require any Tivoli Data Warehouse software or DB2 Warehouse components. You can choose to install a warehouse agent on the same server containing the central data warehouse to improve the performance of data transfer between warehouse databases (refer to 3.2.8, “Warehouse agents” on page 61).

Tivoli Data Warehouse V1.3 will support up to four central data warehouse databases. You can have one or more central data warehouse databases in a Tivoli Data Warehouse deployment on Windows and UNIX. On z/OS, you can have *only one* central data warehouse database (for example, see Figure 3-7 on page 60). The central databases can be installed in the following configurations:

- Only one central data warehouse database on Windows and UNIX platforms *or* one central data warehouse database on a z/OS system

- Up to four central data warehouse databases on Windows and UNIX platforms
- Up to three central data warehouse databases on Windows and UNIX platforms *and* one central data warehouse database on a z/OS system

> **Note:** If you plan to install warehouse packs that were created to run on Tivoli Enterprise Data Warehouse V1.1, you need to create at least one central data warehouse database on a Windows or UNIX system. These warehouse packs use only the first central data warehouse database that is created on a Windows or UNIX system. This central data warehouse database must be named TWH_CDW.
>
> A central data warehouse database on a z/OS system can be populated only by warehouse packs developed for Tivoli Data Warehouse V1.3.

On Windows and UNIX platforms, the first central data warehouse database created by Tivoli Data Warehouse V1.3 is named TWH_CDW. Subsequent central data warehouse database will be named TCDW1, TCDW2, and TCDW3.

On z/OS systems, there are no hard specifications for the name used for the central data warehouse database. However, it is a good practice to follow the naming convention adopted by Tivoli Data Warehouse V1.3.

Multiple central data warehouse databases might be useful in the following situations:

- If your Tivoli Data Warehouse deployment contains systems in widely separated time zones or geographies. A central data warehouse ETL typically runs during off-peak hours to avoid impacting the performance of your operational data stores: Having central data warehouse databases located on servers in different time zones enables you to schedule ETLs for each system at an appropriate off-peak time.
- If your deployment includes z/OS systems. Warehouse packs that use data extracted from z/OS data sources must load their data into a data mart database on a z/OS system. In contrast, warehouse packs that use operational data stores from Windows or UNIX systems can load that data into a data mart database on any supported operating system. Therefore, when you have sources on z/OS and on distributed systems, you must have at least one central data warehouse database on a z/OS system (see Figure 3-5 on page 58 and Figure 3-6 on page 59).

  You can optionally choose to have a second central data warehouse database on a distributed system in order to keep distributed applications data completely separate from z/OS applications data (see Figure 3-7 on page 60).

- If you want to distribute the central data warehouse workload. When using different warehouse packs that do not provide cross-application reports, you can have each warehouse pack load its data into separate central data warehouse databases. This enables you to schedule the central data warehouse ETLs for both warehouse packs to run at the same off-peak time without causing database performance problems.

When planning to use multiple central data warehouse databases, consider the following information:

- If you use a set of warehouse packs that collect historical data intended for the same reporting purpose, all of the warehouse packs must write their data into the same central data warehouse database.

  Note that if a warehouse pack supports extracting data from multiple central data warehouse databases, its documentation contains information about the placement of the central data warehouse databases.
- Distributed application data can flow through a central data warehouse database either on a z/OS or on a distributed system into a data mart database either on a z/OS or on a distributed system.
- z/OS application data can flow only through a central data warehouse database on a z/OS system into a data mart database on a z/OS system.

Important: Although it is possible for a data analysis program to read data directly from central data warehouse databases without using data marts, we do *not* recommend this and it is not supported.

Analyzing historical data directly from the central data warehouse database can cause performance problems for all applications using the central data warehouse.

3.2.4 Data marts

The data marts for your enterprise are contained in a separate set of IBM DB2 databases. Each data mart contains a subset of the historical data from the central data warehouse that satisfies the analysis and reporting needs of a specific department, team, customer, or application.

With Tivoli Data Warehouse V1.3, there are up to four data mart databases supported. You can have one or more central data warehouse databases in a Tivoli Data Warehouse deployment on Windows and UNIX. You can have one or more data mart databases for each central data warehouse deployed on Windows, UNIX, and z/OS systems.

Archived

On z/OS systems, Tivoli Data Warehouse V1.3 only supports one data mart database per IBM DB2 subsystem. In addition to that, the central data warehouse and data marts must be in the same IBM DB2 subsystem, and the IBM DB2 subsystem must have an unique location name.

On Windows and UNIX platforms, the first data mart database created by Tivoli Data Warehouse V1.3 is named TWH_MART. Subsequent central data warehouse database will be named TMART1, TMART2, and TMART3.

On z/OS systems, there are no hard specifications on the name for the data mart database. However, it is a good practice to follow the naming convention adopted by Tivoli Data Warehouse V1.3.

Each data mart database can contain the data from multiple central data warehouse databases.

The data mart databases do not require any Tivoli Data Warehouse software or DB2 Warehouse components, but you can choose to install a warehouse agent on the servers containing the data mart databases to improve the performance of data transfer from central data warehouse databases (refer to 3.2.8, “Warehouse agents” on page 61)

Note: If you plan to install warehouse packs that were created to run on Tivoli Enterprise Data Warehouse V1.1, you need to create at least one data mart database on a Windows or UNIX system. These warehouse packs use *only* the first data mart database that is created on a Windows or UNIX system (TWH_MART).

A data mart database on a z/OS system can be populated *only* by warehouse packs for Tivoli Data Warehouse V1.3.

Multiple data mart databases might be useful in the following situations:

- If your deployment includes z/OS systems. Warehouse packs that use data extracted from z/OS data sources must load their data into a data mart database on a z/OS system. In contrast, warehouse packs that use operational data stores from Windows or UNIX systems can load that data into a data mart database on any supported operating system. You might optionally place a data mart database on a Windows or UNIX system to keep data from those systems separate from data from z/OS applications (see Figure 3-6 on page 59).

- If you want to store your enterprise data in different database for security reasons. You can allow each user to access only the data mart database containing the information which that user is authorized to examine.
- If you plan to access data marts using different reporting or data analysis programs. You can format the data and you can tune each data mart database according to the program that is used to analyze it and the expected workload.

For an effective planning of data mart databases locations, you should consider these requirements:

- Each warehouse pack provides its own data structure called star schema.
- A single data mart database can contain many star schemas.
- Data from different warehouse packs can be stored in the same data mart database, each using its separate star schema.
- Each warehouse pack can write to only one data mart database, and it must pull all of the data for the data mart from a single central data warehouse database.
- Different star schemas in one data mart database can pull their data from different central data warehouse databases.
- Data mart databases on a Windows or UNIX system cannot pull z/OS applications data while a data mart database on z/OS can receive data coming all supported platforms (z/OS, UNIX, and Windows).

3.2.5 Single machine installation

You can install all components of Tivoli Data Warehouse on a single machine (see Figure 3-1 on page 54). This configuration uses the quick deployment method. It is easy to set up and to maintain but is recommended only for test or demonstration environments. It is also useful for those who do not plan to have many users concurrently accessing the data stored in the Tivoli Data Warehouse database, those who do not need to capture and analyze large amounts of data, or both.

The control server can be on Windows NT, Windows 2000 Server, Windows 2000 Advanced Server, or Windows Server 2003. The single machine configuration is not available on UNIX z/OS systems.

Archived

Important: In all the following figures about possible installation architectures, we assume that you are using Crystal server under a Web server. In this case, the only possible installation for the version of Crystal Enterprise 10 that is shipped with Tivoli Data Warehouse is on Windows. However, you do have the option of installing Crystal server on AIX, Sun, and Linux as well, but that requires a servlet engine such as WebSphere or Tomcat. In the following figures, we assume that you are not using a servlet engine and are thus installing on Windows.

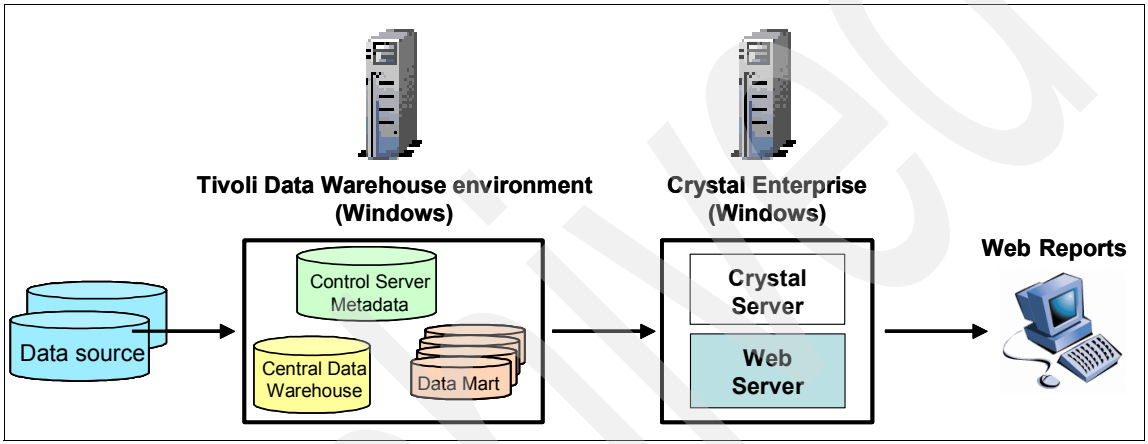


Figure 3-1 Single machine installation

3.2.6 Distributed deployment on UNIX and Windows servers

Most production environments usually require that the Tivoli Data Warehouse components be installed on different servers in order to distribute workload or leverage preexisting infrastructures.

Next, we consider some typical scenarios for UNIX and Windows environments:

- All Tivoli Data Warehouse components on different Windows or UNIX servers:

  The control server is on a Microsoft Windows or AIX server, a central data warehouse database and data mart database on separate large server-class computers (Windows or UNIX), and Crystal Enterprise on a Microsoft Windows or AIX server, as shown in Figure 3-2 on page 55.

Archived

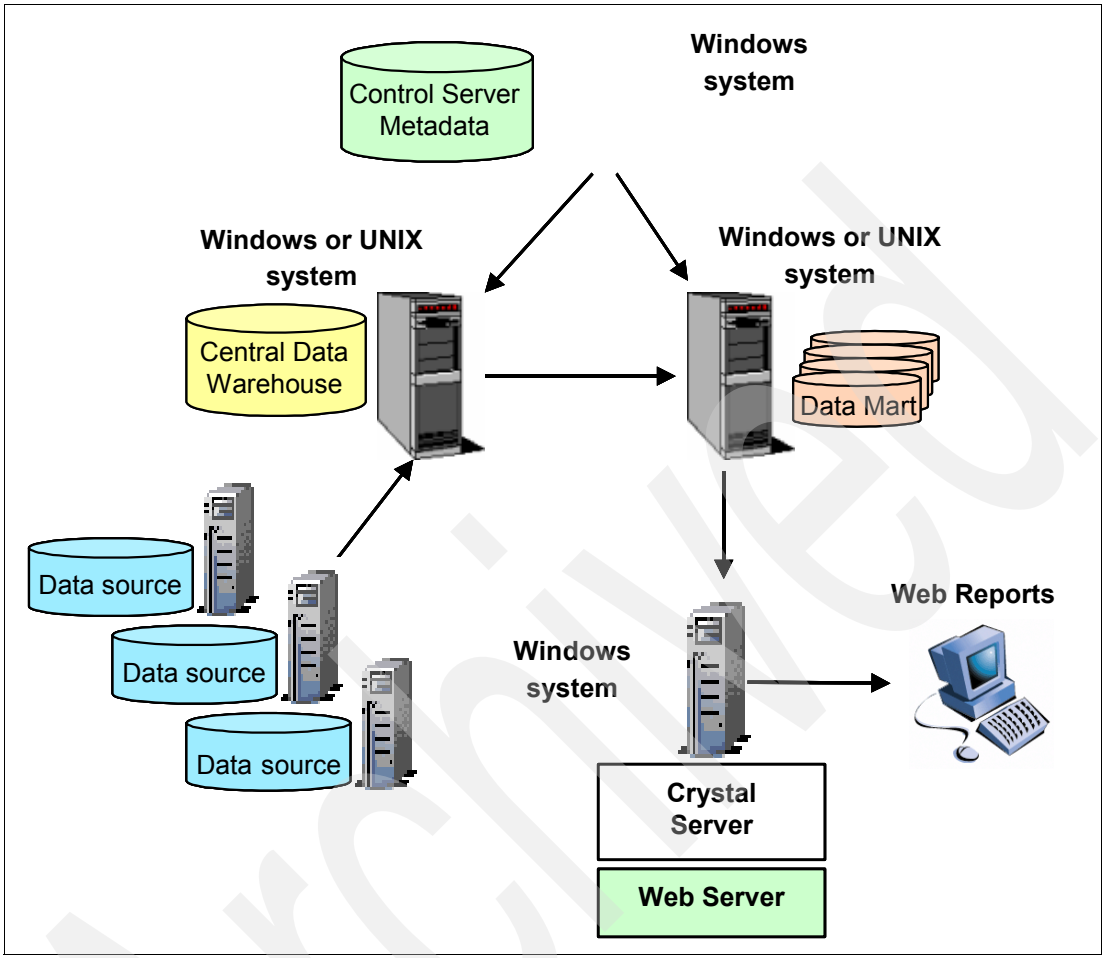


Figure 3-2 Distributed deployment on Windows and UNIX systems

- Central data warehouse and source databases on the same server:

  The central data warehouse database is on the same computer as the database containing the operational data sources. The control server and Crystal Enterprise are on two different Microsoft Windows or AIX servers, as shown in Figure 3-3 on page 56.

  Because operational data sources usually have a high rate of transactions per hour, we do not recommend sharing the same IBM DB2 server for data source and data marts: This configuration might increase the time needed to obtain reports from data marts.

Archived

However, it is possible to have a common IBM DB2 server for data sources and central data warehouse without affecting the performance whenever the ETL1 and ETL2 can be scheduled in off-peak times.

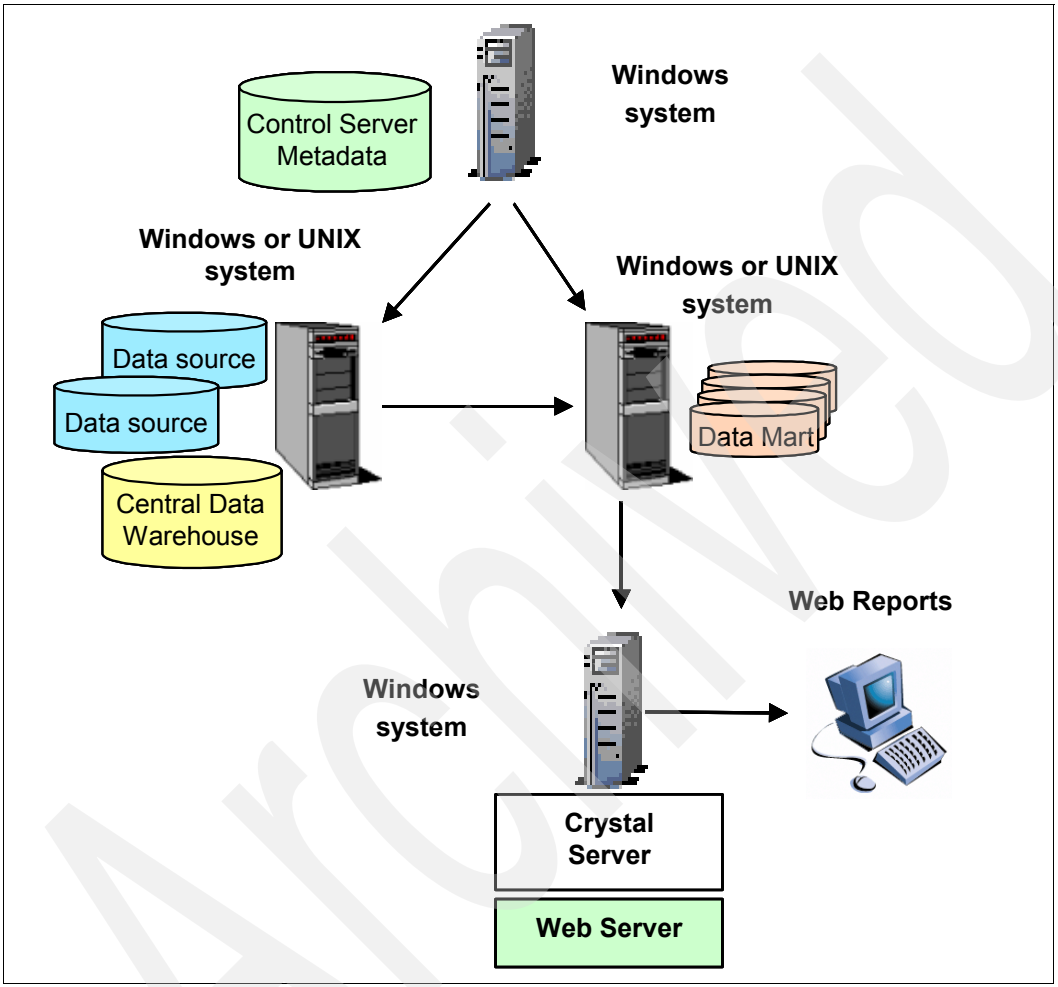


Figure 3-3 Operational data sources and central data warehouse databases on the same server

Archived

3.2.7 Distributed deployment on z/OS, UNIX, and Windows servers

Now, we describe some examples of Tivoli Data Warehouse V1.3 architecture design scenarios that also include z/OS system management data sources:

- Data sources, central data warehouse, and data mart database on a z/OS system:

  The central data warehouse and the data mart database are in an IBM DB2 UDB for OS/390 and z/OS. The control server and Crystal Enterprise are on two different Microsoft Windows or AIX servers, as shown in Figure 3-4.

  This kind of configuration is typically used when all the management data source comes from z/OS applications. The reason is that all warehouse packs extracting data from z/OS data sources must load their data into a central data database located at the z/OS system.

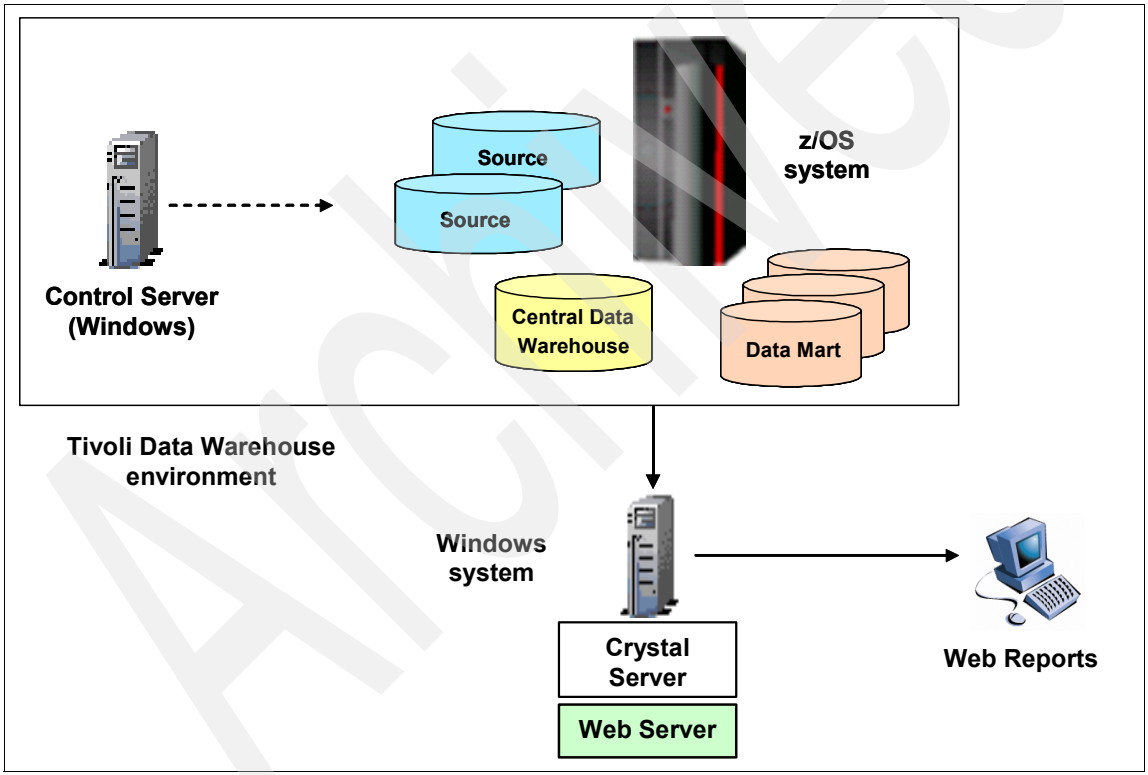


Figure 3-4 Data sources, central data warehouse, and data mart databases on a z/OS system

Archived

- Operational data sources both on z/OS and on distributed systems:

  You can transfer data to a central data warehouse database on a z/OS system even if your operational data sources are distributed among z/OS and distributed systems, as shown in Figure 3-5. In this configuration, you can use only warehouse packs for Tivoli Data Warehouse V1.3, because the warehouse packs for Version 1.1 do not allow any data transfer to a central data warehouse located on a z/OS system.

  The common data mart database on z/OS provides the reports for both data sources on z/OS and distributed systems.

  You can choose to have the common data mart database on a distributed system instead of z/OS, but in that case, you cannot have any reports from operational data sources on z/OS, as shown in Figure 3-6 on page 59.

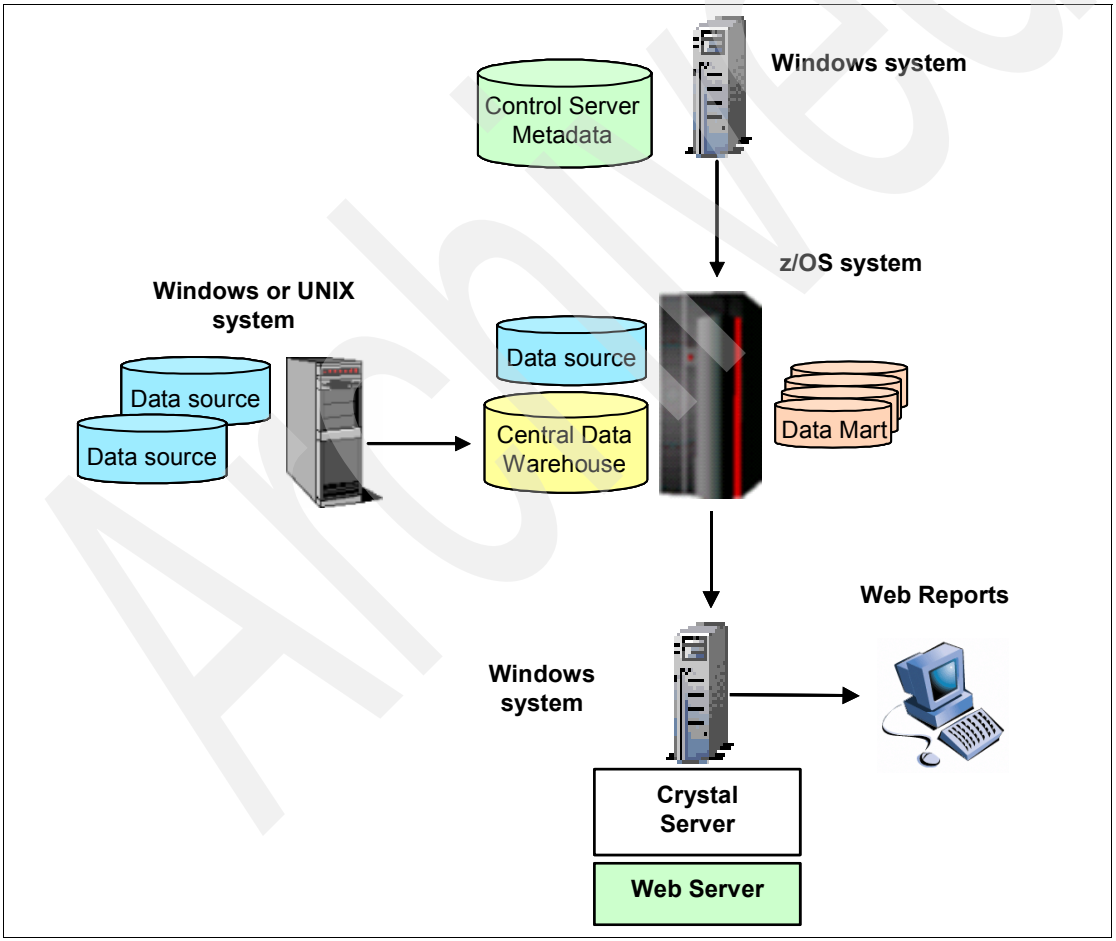


Figure 3-5 Operational data sources both on z/OS and on distributed systems

- Separate data mart databases on a z/OS and on a distributed system:

  This configuration is typically used to segment the reporting functions into two logical areas, one for z/OS and the other for the distributed environment.

  All z/OS application data flows through the central data warehouse database to the data mart database on z/OS, while the distributed applications data is transferred to a data mart on a distributed system. See Figure 3-6.

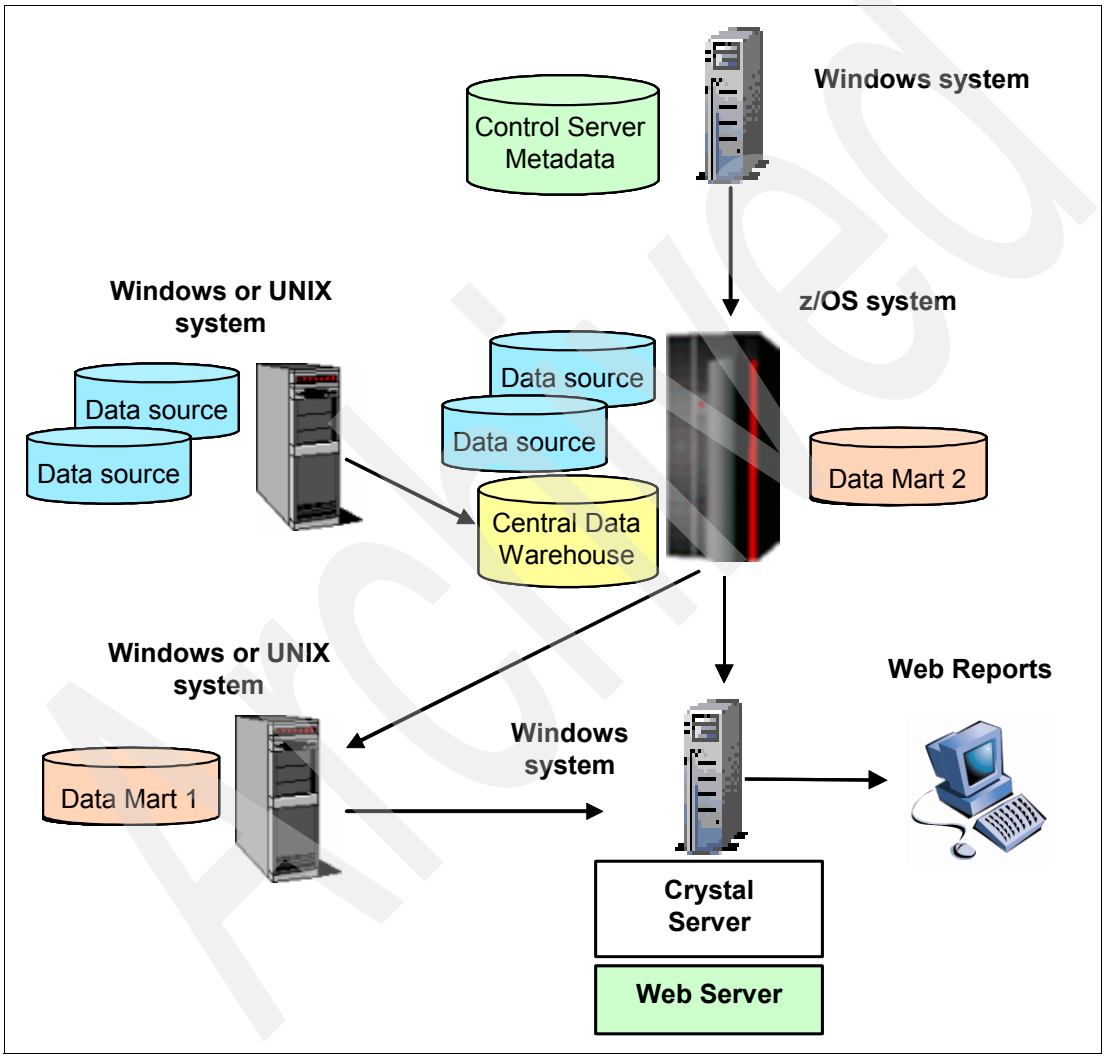


Figure 3-6 Separate data mart databases on z/OS system and distributed system

Archived

- Two central data warehouse servers, one on z/OS, and one on a UNIX or Windows system:

  This is a more complex deployment with one central data warehouse and one data mart database in DB2 UDB for OS/390 and z/OS, a second central data warehouse database on a UNIX or Windows server, and the control server and Crystal Enterprise server on separate Windows systems, as shown in Figure 3-7.

  This configuration might be chosen by customers who want to keep completely separate z/OS applications data from distributed applications data.

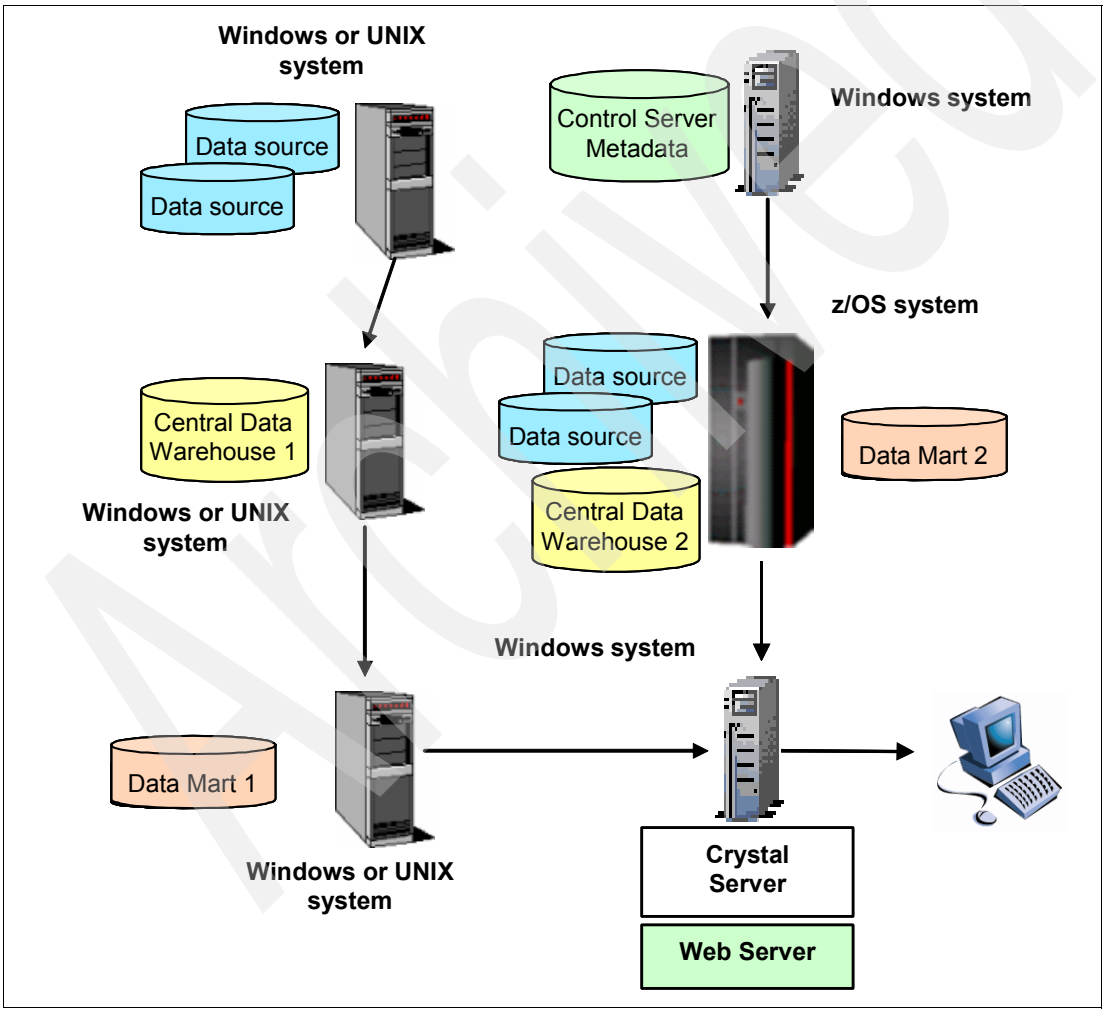


Figure 3-7 Two central data warehouses on a Windows or UNIX system and on a z/OS system

Archived

3.2.8 Warehouse agents

Warehouse agents are IBM DB2 Warehouse Manager components that manage the flow of data between warehouse sources and targets using DB2 CLI or Open Database Connectivity (ODBC) drivers to communicate with different databases.

Every system having a warehouse agent installed is called an *agent site* or a *remote agent*. Tivoli Data Warehouse supports warehouse agents on Windows and UNIX operating systems, but not on z/OS.

When you install the Tivoli Data Warehouse control server, a warehouse agent is automatically installed on the control server machine. In the basic configuration, shown in Figure 3-8, the control server uses its local warehouse agent to manage data flow from the operational data sources to the central data warehouse (ETL1 process) and from that to the data marts (ETL2 process). If the Tivoli Data Warehouse databases are located on the same system as the control server, the warehouse agent is not used.

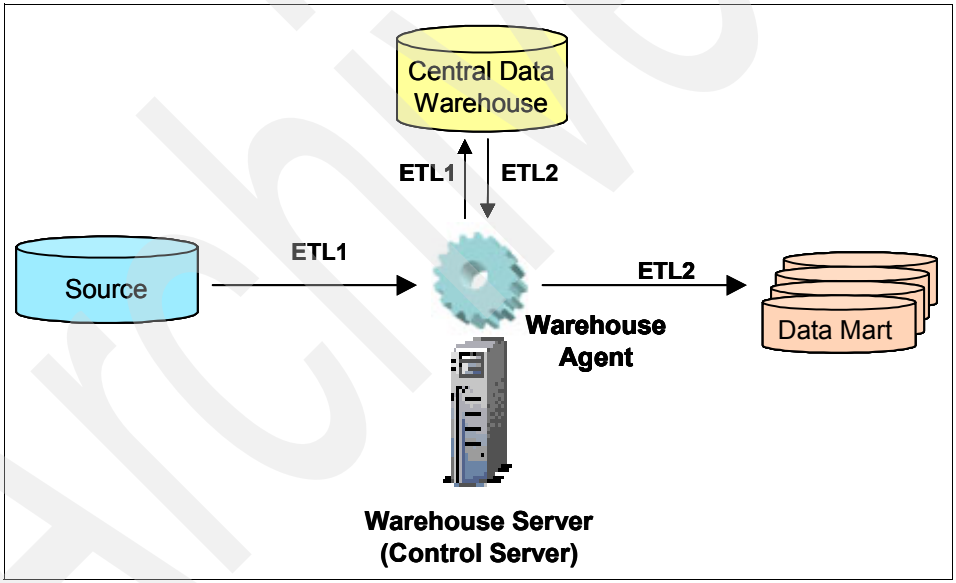


Figure 3-8 Warehouse agent on control server

In a distributed scenario, as shown in Figure 3-9 on page 62, you might improve the performance of Tivoli Data Warehouse by placing a warehouse agent on each central data warehouse server and data mart server. These remote warehouse agents allow a straight data flow from the target to source without passing through the control server, reducing the workload on this server and increasing the speed of data transfer.

Archived

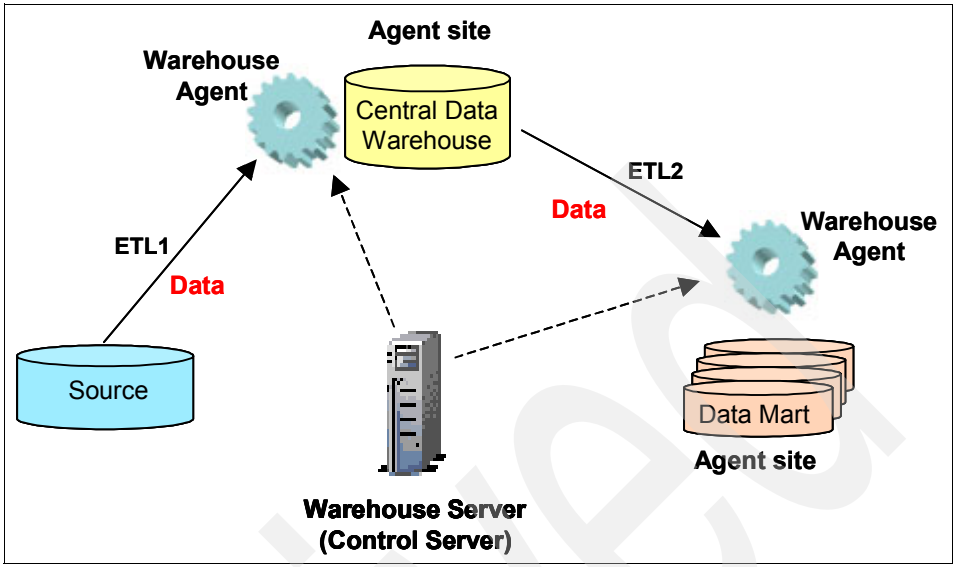


Figure 3-9 Warehouse agents on data targets

Typically, the warehouse agent is placed on the *target* of a data transfer. In this configuration, the warehouse agent performs the following tasks:

- Passes SQL statements that extract data from the remote source tables
- Transforms the data if required
- Writes the data to the target table on the local database

This configuration offers the best performance in a distributed environment by optimizing the DB2 data flow and using block fetching for the extraction. This is the recommended configuration to use with Tivoli Data Warehouse when the source and target are on physically separate machines.

The warehouse agent can be installed on the system that contains the source database. In this configuration, the agent:

- Passes SQL statements that extract data from the local source tables
- Transforms the data if required
- Writes the data to the target table on the remote database

This alternative configuration does not optimize the DB2 data flow (Distributed Relational Database Architecture, DRDA®, or DB2 private protocols) and should be used only if justified by specific architecture requirements. For example, if your data source and data mart databases are on distributed systems while the central data warehouse database is on a z/OS system, you are forced to place warehouse agents on the distributed systems, one of which is the source of data transfer, as shown in Figure 3-10 on page 63.

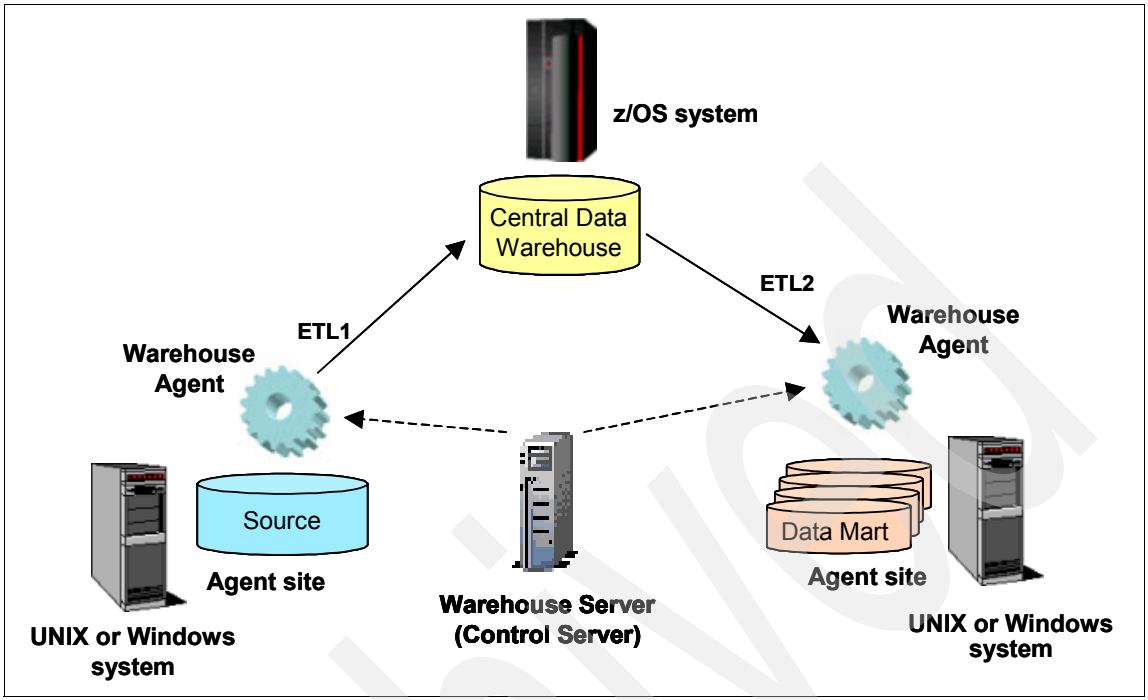


Figure 3-10 Configuration with a warehouse agent on the source

To improve the performance of your ETL process, you should carefully plan where to place agent sites in your environment and which site to associate with each ETL. Table 3-6 suggests where to place the warehouse agents when transferring data from data sources to the central data warehouse database in different scenarios.

Table 3-6 Where to place agent sites for data transfers to a central data warehouse

Operational data source location	Central data warehouse database location	Warehouse agent location
Windows or UNIX system	A different Windows or UNIX system	Central data warehouse system
	The same system	No agent required
	A z/OS system	Operational data source system

Archived

Operational data source location	Central data warehouse database location	Warehouse agent location
z/OS system	The same z/OS location	No agent required
	A different z/OS location	Control server
	A Windows or UNIX system	Deployment not supported. Data sources on z/OS can load data only in a central data warehouse on z/OS.

Table 3-7 suggests where to place the warehouse agents when transferring data from central data warehouse databases to data mart databases in different scenarios.

Table 3-7 Where to place agent sites for data transfers to data marts

Central data warehouse database location	Data mart database location	Warehouse agent location
Windows or UNIX system	A different Windows or UNIX system	Data mart system
	The same system	No agent required
	A z/OS system	Central data warehouse system
z/OS system	The same z/OS location	No agent required
	A different z/OS location	Control server
	A Windows or UNIX system	Data mart

Note that Tivoli Data Warehouse automatically recognizes when the source and target data are on the same computer, and in that case, it transfers the data without using the warehouse agent.

Here are some common situations in which data transfer does not use the warehouse agent:

- When using operational data sources, central data warehouse, and data mart in the same IBM DB2 location on a z/OS system
- When the operational data and the central data warehouse database are on the same computer running Windows or UNIX
- When transferring data between a central data warehouse database and data mart database on the same computer running Windows or UNIX

To install warehouse agents, you must install IBM DB2 Warehouse Manager and Tivoli Data Warehouse on each machine that will be an agent site. If you are using operational data stored in databases other than IBM DB2 (Oracle, Informix, and so on), you are also required to install on that computer a database client for each type of remote database that the agent needs to access. For example, if the operational data source for a warehouse pack is an Oracle database on another computer, you must install also an Oracle database client on the agent site.

For more information about warehouse agents, refer to the *IBM DB2 Warehouse Manager Standard Edition Installation Guide*, GC27-1122, and the IBM Redbook *DB2 Warehouse Management: High Availability and Problem Determination Guide*, SG24-6544.

3.2.9 Considerations about warehouse databases on z/OS

As already explained in the previous sections, there are some limitations when placing Tivoli Data Warehouse databases on z/OS. Here is a summary:

- Only one central data warehouse database can be placed on a z/OS system.
- Data from applications on z/OS or OS/390® systems can be placed only in central data warehouse and data mart databases in DB2 UDB for OS/390 and z/OS.
- To extract operational data from a Tivoli Decision Support for OS/390 database, the central data warehouse database and the Tivoli Decision Support for OS/390 database must be in the same DB2 subsystem.
- The central data warehouse database and data mart database can be a storage group that is shared with other applications.
- Only one data mart can be installed per single DB2/390 location (same host and port).
- Tivoli Data Warehouse V1.3 does not support warehouse agents on z/OS systems.

Before installing a Tivoli Data Warehouse database on a z/OS system, you should know the following parameters about the existing IBM DB2 installation:

- Host name of the z/OS system
- TCP/IP port configured for IBM DB2
- Database location
- Storage volume, storage group, and storage VCAT
- Buffer pool
- Tablespace name
- UTF8 tablespace name
- Tablespace size (primary and secondary)

Archived

3.2.10 Coexistence with other products

This section describes coexistence issues between Tivoli Data Warehouse and other applications such as Tivoli Decision Support (distributed), Tivoli Decision Support for OS/390, and other IBM DB2 databases and data warehouses:

- Coexistence with Tivoli Decision Support:

  You can continue to use Tivoli Decision Support at the same time and on the same systems as Tivoli Data Warehouse. For the best performance, do not schedule Tivoli Data Warehouse ETLs and Tivoli Decision Support cube builds concurrently if they access the same databases.

- Coexistence with Tivoli Decision Support for OS/390:

  You can continue to use Tivoli Decision Support for OS/390 at the same time and on the same systems as Tivoli Data Warehouse. For the best performance, schedule Tivoli Data Warehouse ETLs after Tivoli Decision Support for OS/390 data collection has occurred.

- Coexistence with other data warehouses:

  Do not mix components from different deployments of Tivoli Data Warehouse on any computer. For example, if you can have both a production deployment and a test deployment, you must not place any warehouse components of the test deployment on the same computer as any components of the production deployment.

  You can use the IBM DB2 Data Warehouse Center to manage multiple data warehouses. However, only one data warehouse can be active (running scheduled jobs) at a time. Scheduled jobs run only for the warehouse whose control database (TWH_MD in case of Tivoli Data Warehouse) is selected in the IBM DB2 Data Warehouse Center. Other warehouses, each identified by a different control database, retain their information but do not run scheduled jobs until their control database is selected in the IBM DB2 Data Warehouse Center.

- Coexistence with other IBM DB2 database applications:

  You can have other applications using the same IBM DB2 instances used by Tivoli Data Warehouse, provided that they do not use the same database names required by warehouse components (TWH_MD, TWH_CDW, TCDW1, TCDW2, TCDW3, TWH_MART, TMART1, TMART2, TMART3, and any other database names defined on z/OS systems).

  If you install Tivoli Data Warehouse using an existing IBM DB2 instance, all other clients that are connected to that instance lose their IBM DB2 connections during the installation process. Therefore, remember to stop all existing connection before starting the Tivoli Data Warehouse installation.

Archived

3.2.11 Selecting port numbers

You must allocate port numbers for Tivoli Data Warehouse for communication between the Tivoli Data Warehouse control server and the remote IBM DB2 databases (data sources, central data warehouse, data marts) and for communications with Crystal Enterprise.

Table 3-8 lists the default port numbers that are used by Tivoli Data Warehouse. You can use the same or different port numbers for the control server, data mart database computers, and central data warehouse database computers.

Table 3-8 Default ports used in Tivoli Data Warehouse environments

Component	Default port
DB2 databases on UNIX and Windows	50000
DB2 databases on z/OS	5000
Crystal Enterprise server	6400
Internal communication between the Crystal APS and other Crystal components, such as the Crystal Web Connector running on the HTTP Server	6401
HTTP server for Crystal Enterprise	80

3.3 Database sizing

A correct database sizing is one of the most important issues you have to face when planning a Tivoli Data Warehouse deployment.

The growth rate of your central data warehouse and data mart databases is related to the number of measured items in your environment (hosts, database instances, events, and so on), the measurements you require for each item, and the sampling frequency.

You can find detailed instructions for estimating the amount of required storage in the implementation guide of each warehouse pack you install in your environment. If the implementation guides do not provide this kind of information, refer to Chapter 4, “Sizing Tivoli Data Warehouse V1.3” on page 81. In this chapter, we illustrate some calculations for estimating the central data warehouse database storage requirements for a warehouse pack.

Archived

3.4 Security

The following sections describe security considerations for installing and using Tivoli Data Warehouse.

3.4.1 Authority required to install and maintain IBM DB2 UDB

For UNIX machines, the user must be logged in with root authority. User and group accounts will be created for the instance owner, Administration Server, fenced user defined functions, and fenced stored procedures. We recommend that you use different user IDs for UDFs and the instance owner.

For Windows machines, the user must be part of the administrators group. Specifically, the user must have these advanced user rights:

- Act as part of the operating system
- Create token object
- Increase quotas
- Replace a process-level token

More detailed user account information can be obtained from the *IBM DB2 UDB Quick Beginnings for DB2 Servers V8.2*, GC09-4836.

3.4.2 Authority required to install Tivoli Data Warehouse

On Windows systems, to install Tivoli Data Warehouse or to install, start, or stop the Crystal Enterprise Professional for Tivoli server, you must be logged on as a locally defined administrator.

On UNIX-based systems, to install Tivoli Data Warehouse, you must be logged in as the root user.

On z/OS systems, to add or remove a central data warehouse database or data mart, SYSADM authority is required, and DBADM authority on central data warehouse and data mart databases is required to install warehouse packs.

3.4.3 Firewalls

You cannot have a firewall between any Tivoli Data Warehouse components. This includes the control server and the Tivoli Data Warehouse agent sites.

However, it is possible to have configurations with a firewall between the source databases and a central data warehouse, or a firewall between a central data warehouse and data marts, if the source database vendor supports communication through firewalls and the Data Warehouse agent resides on the

central data warehouse. Figure 3-11 shows both a valid configuration and a non-functional configuration. Only the ODBC communication can be transported through the firewall.

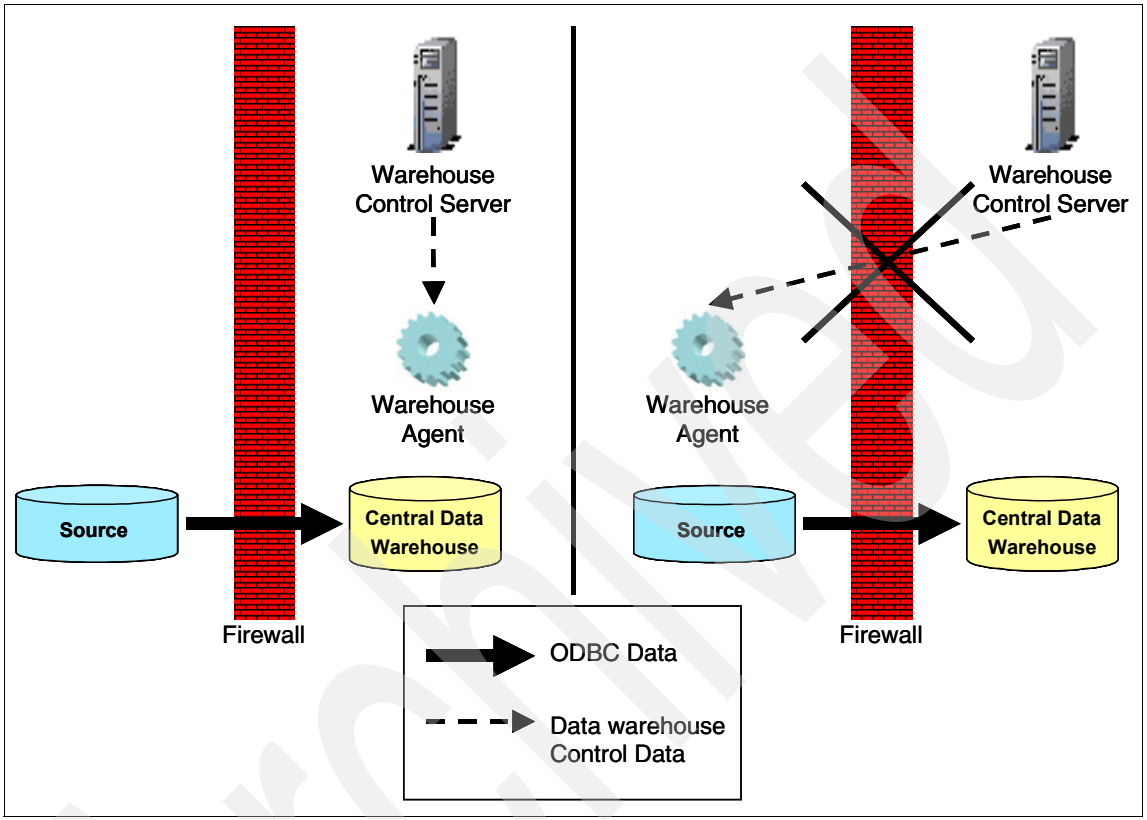


Figure 3-11 Tivoli Data Warehouse and firewalls

Crystal Enterprise can deliver a broad range of reporting and analytic content to any browser using pure DHTML. Unlike plug-in based technologies, DHTML requires no software downloads and no special configuration to enable viewing, making it ideal for deployments through a firewall without compromising security.

3.4.4 Controlling access to data in the warehouse

The following techniques can be used to control access to data in Tivoli Enterprise Data Warehouse:

- Restricting which DB2 users have access to specific views in the central data warehouse and in the data mart database. This is done using DB2 administrative tools.

- Using multicustomer or multicenter support, as described in 3.4.6, “Multicustomer and multicenter support” on page 71.

3.4.5 Protecting information in Crystal Enterprise Professional for Tivoli

Crystal Enterprise Professional for Tivoli enables you to choose one of three types of user authentications:

- Enterprise
- Windows NT
- LDAP

Enterprise authentication is based on Crystal proprietary, internally defined accounts and groups used only for Crystal Enterprise.

Windows NT authentication uses already existing Windows NT and 2000 user accounts and groups for authentication in Crystal Enterprise Professional for Tivoli, eliminating the need to create new additional accounts within the application.

LDAP authentication can be chosen if you have already an LDAP directory server in your environment. In this case, you can use existing LDAP user accounts and groups also for Crystal Enterprise Professional for Tivoli.

Crystal Enterprise Professional for Tivoli not only authenticates its users, but also allows an access control to all the objects within the application using the *Crystal Management Console* (CMC). The base unit for controlling consists of specific object rights that provide a user or a group with permission to perform a particular action on an object. Each object right can be Explicitly Granted, Explicitly Denied, or Not Specified.

The Crystal Enterprise Professional for Tivoli object security model is designed according a “denial base” in order to guarantee that users and groups do not automatically acquire rights that are not explicitly granted. All rights that are not specified are denied by default. Additionally, in case of contradictory settings that grant and deny the same right to the same user or group, the right is denied by default.

Crystal Enterprise includes a set of predefined access levels that enable you to set common security levels quickly and facilitate administration and maintenance. Each access level grants a set of rights that combine to enable users to accomplish common tasks such as viewing reports and scheduling reports.

Users can inherit rights as the result of group membership, subgroups can inherit rights from parent groups, and both users and groups can inherit rights from parent folders. However, you can always disable inheritance or customize security levels for particular objects, users, or groups.

Refer to the manual, *Crystal Enterprise Professional Version 9 for Tivoli Administrator's Guide*, which is provided with Crystal Enterprise Professional Version 9 for Tivoli, for further information about security aspects of Crystal Enterprise Professional for Tivoli.

3.4.6 Multicustomer and multicenter support

Tivoli Data Warehouse enables you to keep data about multiple customers and centers in one central data warehouse, while restricting access so that customers can see and work with data and reports based only on their data and not any other customer's data.

Important: If you intend to have multicustomer and multicenter support in your Tivoli Data Warehouse, you must plan and configure for this before importing any data into your central data warehouse.

If you decide to enable multicustomer and multicenter support after you have already imported some data, you must set up a new central data warehouse for the new multicustomer environment and, if necessary, migrate your old data into the new environment.

In order to support a multicustomer environment, you must be sure that your application data correctly discriminates between different customers. Various applications can use different data fields to distinguish between customers, but within applications, you should make the data fields consistent.

You must determine which field in a data source is applicable for identifying multiple customer data. An application might use a single data field to determine customer uniqueness, while other applications might use two or more data fields to distinguish customer data. Either case is permitted as long as the data fields used are consistent within a single application.

When the central data warehouse ETL is run, data from applications is assigned valid customer account codes by matching certain data fields in the incoming data with pre-identified values in a matching database table. Because each application can use different fields and different numbers of fields to identify customers, each application has its own matching table that it uses during the central data warehouse ETL process.

Archived

To configure the central data warehouse for multicustomer functionality, you must manually define your customers into the TWG.CUST and *Product_Code*.CUST_LOOKUP tables, where *Product_Code* is the code univocally associated with the warehouse pack you intend to use. In the same way, you can configure the multicenter support by simply defining your centers into the TWG.CENTR and *Product_Code*.CENTR_LOOKUP tables on the central data warehouse database.

3.5 Network traffic considerations

An accurate planning of a Tivoli Data Warehouse environment should also consider the possible impact on the network.

When the ETLs run, large amounts of data are transferred, and this can greatly affect the network performance. However, a correct scheduling of the ETLs and a proper design of Tivoli Data Warehouse implementation can reduce the impact on the network.

> **Note:** The ETLs are based on Structured Query Language (SQL). The SQL processing exhibits a typical request/response pattern, meaning that requests send small amounts of data, while responses return large amounts of data. This can have an adverse effect on the network.

Every customer environment will have unique network requirements. Therefore, Tivoli Data Warehouse must be tailored to fit each customer's specific needs. In this section, we discuss some of the decisions that a customer will face when planning for Tivoli Data Warehouse.

3.5.1 Architectural choices

The single machine installation of Tivoli Data Warehouse (Figure 3-1 on page 54) affects the network far less than any other distributed installation (for example, see Figure 3-2 on page 55). Simply put, there is less data transferred across the network in a stand-alone environment than on a distributed environment.

However, a single server cannot handle the entire workload of a Tivoli Data Warehouse for most production environments, and then a distributed installation is very often required. In this case, the network performance can be improved by using remote warehouse agents and carefully placing the data marts and the Crystal Enterprise Professional for Tivoli server.

Remote warehouse agents

During a typical distributed installation, a local warehouse agent is installed on the system with the control server. Using the local agent is the easiest and fastest in terms of installation, but this configuration is the least efficient in terms of performance and network traffic.

In an advanced distributed Tivoli Data Warehouse, remote warehouse agents are usually located on the target databases. In this setup, communication between the source or target and the control server is minimized.

Refer to 3.2.8, “Warehouse agents” on page 61 for more details about remote warehouse agents and their possible locations.

Data marts and Crystal Enterprise server

Crystal Enterprise Professional for Tivoli server uses an IBM DB2 client connection to retrieve data from data mart databases. You can reduce the network traffic by simply installing the data mart databases on the same machine where the Crystal Enterprise Professional for Tivoli server is located.

However, this configuration requires that the data mart databases be on a Windows server, while you might prefer to have them on a UNIX or z/OS system.

3.5.2 Scheduling

As stated earlier, ETLs transfer potentially large amounts of data over the network. In a production environment, it is advised not to run these during normal business hours. Instead, ETLs should be scheduled to run when network traffic is low.

Production versus maintenance windows

The production and maintenance windows should be the first parameters defined when determining when to schedule ETLs. Generally, the production window consists of normal business hours. This usually means the hours representing the highest traffic volume on the network. This is also the time during the day that end users view and run reports against the data marts.

ETLs and database backups should not be run during the production window, because they can both negatively impact report performance and other users across the network as well. In contrast, the maintenance window consists of non-business hours, which represent the off-peak hours on the network. ETLs and backups should run during this time.

If your environment is spread over wide geographical areas, you might choose to have different central data warehouse and databases for different time zones.

Archived

This configuration enables you to schedule ETLs separately during each maintenance window in the different time zones (see 3.2, “Physical and logical design considerations” on page 47).

ETLs

After establishing the production window, the ETLs can now be scheduled. Once again, different strategies can be implemented to fit the needs of the customer. There are a number of different variables that come into play.

Two important variables to consider are the number of warehouse packs and the types of warehouse packs installed in Tivoli Data Warehouse. One or two warehouse packs that extract large amounts of data can take much longer than the combination of many warehouse packs that extract small amounts of data.

ETL grouping

Another consideration when planning the scheduling of ETLs should be grouping. When you have several warehouse packs installed in your environment, you can use two different approaches for the ETLs’ schedule:

- First, run all the ETL1s (from operational data sources to central data warehouse), and then all the ETL2s (from central data warehouse to data mart).
- Run sequentially the ETL1 and ETL2 for each warehouse pack.

The first approach will place most of the network load at the beginning of your maintenance window when ETL1s are run. This is because the ETL1s query the application databases and must transfer data across the network when doing so. ETL2s extract data from the central data warehouse and load it into the data marts. If these two databases reside on the same machine, network traffic is minimal. Therefore, if the total time of all ETLs runs into the production window, there is relatively little impact on the network. The drawback is that if the ETL2s carry over into the production window, the reports will be unavailable during normal business until the ETLs complete.

The second approach guarantees that at least some data marts would be already updated at the start of business even if the total time of all ETLs runs into the production window. The drawback is that an ETL1 might still be running at the start of business, which might put a heavy load on the network at a high peak time.

Tip: ETL execution times are largely dependent on the amount data they query. Always try to schedule ETLs that handle large amounts of data before your ETLs that handle less data in the maintenance window when considering either strategy.

3.6 Integration with other business intelligence tools

Tivoli Data Warehouse V1.3 natively provides Web reports through Crystal Enterprise Professional Version 9 for Tivoli. However, if you have existing business intelligence infrastructures already in place or if you prefer using other tools to analyze your data, you can easily integrate them with Tivoli Data Warehouse.

Note: If you do not want to use Crystal Enterprise Professional for Tivoli, you do not need to install it. You can type in NO_CRYSTAL for the host name on the Crystal panel and it will not require a valid Crystal system to connect to. You also need to type in a user name in this panel, but that can be anything; it is not checked.

You can connect your preferred applications to the data marts and extract your own reports without using Crystal Enterprise Professional for Tivoli. Technically, these applications might report directly from central data warehouse, but this is a non-supported configuration, provides very poor performance, and might interfere with the ETL processes (see Figure 3-12).

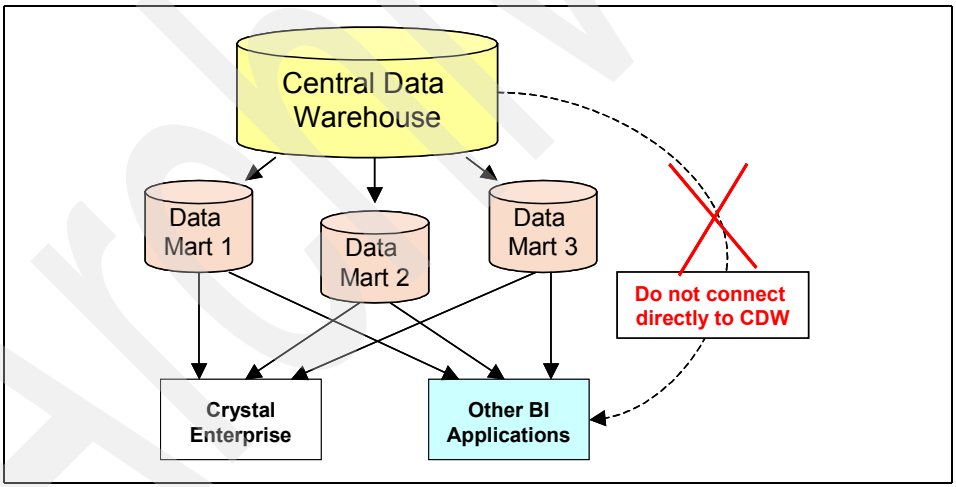


Figure 3-12 Business intelligence integration

By having only a subset of the data in the data mart, the database system can manage the data faster and easier. Also, because the filtering has already been applied at the ETL2 run time, the reporting queries become much smaller.

Smaller queries performing on a small subset of data are, of course, easier to tune. Therefore, the end customer will experience better reporting performance.

Refer to the Redpaper *Tivoli Data Warehouse 1.2 and BusinessObjects*, REDP-9116, for details about how to integrate with business objects. This Redpaper might also give you an idea about how to integrate with other business intelligence tools.

3.7 ETL development

A typical organization generally relies on very different tools to generate data for its business and stores historical information on separate repositories and in a variety of formats, including relational databases, spreadsheet data, and log files. These repositories have different data schemas, because they were designed by different vendors, and they might have metrics specific to that platform. While platform-specific reports are usually required, it is often useful to create enterprise-level reports, which mix information from the various platforms. An example of this is to report performance utilization statistics of servers in a single report, regardless of the platform architecture.

Tivoli Data Warehouse allows mapping of the existing data sources into one or more central data repository the repository so that reports having a common look and feel can be easily generated, thus improving the understanding of different platforms.

In addition to having different data repositories caused by different platforms, different tools are frequently deployed on the same platforms, but on different servers. This is usually the case when these servers are supported by different personnel and then consolidated at some future time (for example, companies that have grown through acquisition, where the servers have come from different companies, or where different support organizations have been allowed to select their own system management tools). By mapping these tools to the central data repository, common reports can be generated, masking the different tools used to create the data.

Lastly, you might want to convert existing infrastructure to Tivoli-based products. By using Tivoli Data Warehouse, historical data from existing servers can be loaded into the central data repository without any data loss. This will enable you to convert over to a Tivoli product that uses the central data repository and not lose any data. While servers are in the process of converting, both the new and old data sources can be used to generate reports in the new common format.

The ETL1 programs take the data from these sources and places it in the central data warehouse, while the ETL2 programs extract from the central data

warehouse a subset of historical data that contains data tailored to and optimized for a specific reporting or analysis task. This subset of data is used to create one or more data marts, a subset of the historical data that satisfies the needs of a specific department, team, or customer.

A data mart is optimized for interactive reporting and data analysis. The format of a data mart is specific to the reporting or analysis tool you plan to use. Customers can then use Crystal Enterprise Professional for Tivoli or other analysis programs to analyze a specific aspect of their enterprise using the data in one or more data marts.

Whenever you need to extract data from sources not supported by existing Tivoli warehouse packs or if you decide to use customized data marts, you have to develop your own ETLs. The guide *Enabling an Application for Tivoli Data Warehouse*, GC32-0745 provides all the information needed to develop customized ETLs.

3.8 Skills required for a Tivoli Data Warehouse project

Implementation of a complete Tivoli Data Warehouse solution requires different skills and usually involves different departments inside a company.

A Tivoli Data Warehouse project can be schematically divided into four phases:

1. Implementation
2. Data collection
3. Data manipulation (ETL1 and ETL2)
4. Reporting

3.8.1 Implementation

The setup of a Tivoli Data Warehouse environment is highly automated and therefore does not require a very specific training.

The *Tivoli Data Warehouse administrator* should have DB2 administrative skills on a distributed platform in order to manage and optimize all the databases used in the data warehouse. If the managed environment also includes z/OS systems, the administrator should have administrative skills in DB2 UDB for OS/390 and z/OS or should be supported by a person with these skills at least during the databases setup on z/OS.

Archived

3.8.2 Data collection

The most important phase of a warehouse project consists of defining all the data sources involved and which is the really relevant data in our business.

Generally, IT departments are responsible for the collection of metrics concerning availability, performance, and utilization of IT resources (systems, applications, and networks). Other departments could provide additional data, such as financial statistics or inventories.

Skills required to select and actually perform data collection vary according the type of data under scope. Generally a *system administrator* or an *application administrator* is directly responsible for the implementation of IT resources monitors or the administrator can provide the information required to implement them.

Tivoli software products already offer a wide range of solutions for collecting data about IT infrastructure, but if a company has already implemented different data collection processes before the Tivoli Data Warehouse installation, there is no need to modify them. The data manipulation phase is in charge of retrieving data from legacy sources and converting it into a common format.

Let us consider the following scenario to explain this further. Suppose that a company plans to store in a Tivoli Data Warehouse all data retrieved by Tivoli monitoring applications, as well as by some preexisting and highly customized applications.

Tivoli Data Warehouse enables that company to correlate data coming in differently from Tivoli and non-Tivoli sources without affecting the preexisting processes: that can be obtained simply customizing the SQL code necessary to transfer data from the old application database to the central data warehouse (source ETL) and that required to populate the data marts (target ETL) that will be used to generate reports.

Note that the ETLs perform not only a plain data transfer between different databases, but they are also in charge of all the transformation tasks required to have a common format for all data independently from the sources.

A typical example of data transformation might be the time stamp of a measurement: Each monitoring application generally uses the local time and that could generate confusion whenever we examine data produced in different areas of the world. Therefore, the ETLs are also required to convert the times referring to a common standard, such as the Greenwich Mean Time.

Another example of data transformation concerns the standardization in the denomination of the measured components: Different applications might use

different names to indicate the same object, and the ETLs must correct any possible mismatch in order to produce always coherent reports when comparing data from different sources.

3.8.3 Data manipulation (ETL1 and ETL2)

After the data sources are established, the next steps are as follows:

1. Loading legacy sources data into central data warehouse and transforming data into a standard format (ETL1 or Source ETL process)
2. Selecting data subsets from a central data warehouse according different business areas, and loading them into data marts (ETL2 process or Target ETL process)
3. Scheduling convenient time frames to run ETL processes

Tivoli already provides free ETLs for most Tivoli products. These ETLs can be immediately used to manipulate data from Tivoli applications, or they can be used as models to create internally developed ETLs to manage other products.

Customized ETLs can be packaged and shipped to customers or colleagues, who can install them using the Tivoli Data Warehouse installation wizard. You can find out how to integrate your own components into the Tivoli Data Warehouse in the manual *Enabling an Application for Tivoli Data Warehouse*, GC32-0745.

Source and target ETLs are developed with the same method and slight variations in design considerations. Skills required to develop ETLs are:

- Standard SQL
- Some experience in data warehousing and fair amount of DB2 skills
- Knowledge of source databases involved

ETL developers should completely understand their end-users' requirements in order to project proper data marts, while each data source administrator is requested to provide ETL developers with the structure of their databases. No interface is provided to allow users unfamiliar with the data and with SQL to simply move data from any application's data store to the data warehouse.

In a Tivoli-only environment with Tivoli warehouse packs, the following skills are recommended:

- Knowledge of the source application
- Knowledge of the Tivoli product used to collect the date (IBM Tivoli Monitoring or IBM Tivoli Storage Manager)

- Basic knowledge of Tivoli Data Warehouse

To set up data collection for a new application, the following skills are needed:

- Knowledge of the source application
- Knowledge of standard SQL
- Knowledge of Tivoli Data Warehouse and its underlying data model

One of the most critical aspects of the data manipulation phase in a large environment is the scheduling of all the different ETL processes running in a Tivoli Data Warehouse. The person responsible for scheduling ETLs should have a thorough knowledge about:

- The timing of data source updates
- The requirements of end users for all reports
- The workload for each server in the Tivoli Data Warehouse environment
- The impact of ETLs on the network

You can find a discussion about the last two items in 3.5, “Network traffic considerations” on page 72.

3.8.4 Reporting

The final step of a Tivoli Data Warehouse process is producing timely, updated reports according different end-users’ specifications.

Tivoli warehouse packs already provide out-of-the-box reports for the Crystal Enterprise Professional for Tivoli Limited Edition bundled with Tivoli Data Warehouse, but if there is a need to define additional customized reports, either Crystal Enterprise Professional for Tivoli Professional Edition or another business intelligence tool is required.

These are the skills required to implement customized reports for Tivoli Data Warehouse:

- Basic knowledge of how to connect the data mart databases through ODBC
- Basic knowledge of standard SQL
- Knowledge of the data mart structure and data
- Experience with Crystal Enterprise products or other business intelligence tools

Report designers usually interact with the ETL2 developers, who are providing all of their requirements about relevant metrics, details about information, aggregation times, preservation of old data, and so on, in order to optimize the star schemas according to their reporting needs.

Archived

4

Sizing Tivoli Data Warehouse V1.3

This chapter discusses sizing a Tivoli Data Warehouse Version 1.3 environment. We address the following issues:

- CPU recommendations and guidelines
- Disk storage requirements
- A sizing example
- The spreadsheet example tool
- Tivoli Data Warehouse V1.3 sizing summary

The key issue in sizing is to remember that Tivoli Data Warehouse V1.3 is a solution built on DB2 Universal Database Version 8.2. Database server hardware is required both for optimal performance and to meet storage requirements for Tivoli Data Warehouse V1.3.

© Copyright IBM Corp. 2005. All rights reserved.

4.1 CPU recommendations and guidelines

Two possible installations represent the most common Tivoli Data Warehouse V1.3 environments. They are:

- A single system install with all Tivoli Data Warehouse V1.3 components on a single server
- A distributed system install with the control server on one machine and the TWH_CDW and TWH_MART databases on a separate database server

In both cases, we recommend that the TWH_CDW and TWH_MART server be dedicated to Tivoli Data Warehouse V1.3 use for optimal performance.

4.1.1 CPUs for a single system installation

Our experience with Tivoli Data Warehouse V1.3 indicates that multiple CPUs can be used during extract, transform, and load SQL scripts (ETL) execution steps. Given the demands on the server hardware in a single system installation, we recommend a two or four CPU server.

Because portions of the ETL execution show 100% utilization of all processors, using a server with the fastest possible processors will give best results. The trade-off between a higher number of CPUs versus a higher CPU processor speed should usually go to the higher CPU speed, as long as a minimum of two CPUs exist in the server. Our recommendation is a four CPU server with the fastest CPU speed available in that configuration.

4.1.2 CPUs for a distributed installation

For a distributed Tivoli Data Warehouse V1.3 installation, the same guidelines follow with one exception.

If the control server is installed on a separate server, it is generally acceptable to use a single processor server or two CPU server. Again, higher processor speed is preferable. The best performance will be available on a two CPU system.

The database server for the TWH_CDW and TWH_MART databases in a distributed installation should be a two or four CPU server with the fastest possible CPUs available. During IBM Tivoli Monitoring Warehouse Pack AMX ETL execution, we observe several time periods when all four processors of a four CPU AIX database server run at or near 100% utilization. If you plan on running this ETL, you should consider a database server with four CPUs available.

4.2 Memory considerations

All servers with Tivoli Data Warehouse V1.3 will be running operating system, DB2 Universal Database Version 8.2, Crystal Enterprise 10 software, or some combination of these. For all servers, a minimum of 2 GB should be the starting point for memory requirements.

For the Tivoli Data Warehouse V1.3 control server and Crystal Enterprise 10 machines, 2 GB is sufficient.

Performance of Tivoli Data Warehouse V1.3 is closely tied to the amount of memory allocated for the TWH_CDW and TWH_MART databases. You should consider moving beyond the 2 GB minimum to as much as 4 GB to 8 GB for the server with these databases. This additional memory will give you the most flexibility in tuning the database performance. See the Chapter 7, “Performance maximization techniques” on page 175for more information.

4.3 Disk storage requirements

In this section, we describe a best practice approach to separate out the components of the databases of Tivoli Data Warehouse V1.3. We start by suggesting a minimal configuration, give some ideas about how to expand that to increase both storage space and performance, and finish by working through a detailed example to demonstrate how to determine the storage requirements for Tivoli Data Warehouse V1.3.

4.3.1 Minimum disk setup

The minimum recommended number of disks for any Tivoli Data Warehouse V1.3 installation is five separate physical devices. This provides a dedicated disk for each of the following components:

Disk 1	Operating system software
Disk 2	DB2 Universal Database Version 8.2 software
Disk 3	TWH_CDW data and indexes
Disk 4	TWH_MART data and indexes
Disk 5	TWH_MD data, indexes, and transaction logs for the TWH_CDW and TWH_MART databases

Separation of the Tivoli Data Warehouse V1.3 components as shown above is recommended and can be accomplished using the database creation scripts included in the samples directory of the Tivoli Data Warehouse V1.3 installation

Archived

media. Moving the transaction logs for the TWH_CDW and TWH_MART databases to a separate physical drive from the data and indexes is a best practice for database performance (see the documentation for details about how to change the transaction logging path using the *newlogpath* database configuration parameter).

For more information about the separation of the Tivoli Data Warehouse V1.3 components on various disks, refer to Chapter 7, “Performance maximization techniques” on page 175.

4.3.2 Going past the minimum disk installation

The previous suggested minimum layout suggests several possibilities for building a more robust environment. One simple approach would be to use logical volumes that stripe data across multiple disks. For example, on a Windows or Linux OS based TWH_CDW database server, a single logical disk can be defined to the operating system that consists of multiple physical hard drives using a RAID disk array. We still recommend separating the components of Tivoli Data Warehouse V1.3, as described in 4.3.1, “Minimum disk setup” on page 83. In an environment that requires more storage space or higher performance, you should follow the suggested layout, but consider creating logical volumes for the TWH_CDW and TWH_MART data and indexes that consist of multiple physical devices with data striped across the disks.

For our distributed installation used for this book, we use an AIX-based database server with multiple disks. We define four disks as a single logical volume and create a JFS file system on that logical volume that was dedicated to the TWH_CDW data and indexes. We do the same for the TWH_MART data and indexes, but use only three disks. We also create a logical volume with data striped across two physical disks for the TWH_CDW and TWH_MART transaction logs.

We strongly recommend that you use one of these approaches. Using multiple disks is better for performance and increases storage capacity.

4.3.3 Storage size requirements

Determining the storage requirements for any Tivoli Data Warehouse V1.3 installation is highly dependent on what warehouse packs are to be installed and how much data you will be pulling into the TWH_CDW and TWH_MART databases.

Ultimately, the primary indicator of how much space will be required for the TWH_CDW database is the number of rows of data that will be stored in the TWG.MSMT table; this is true for all warehouse packs except for the Tivoli

Enterprise Console Version 3.9 warehouse pack (see the documentation for this warehouse pack for details).

The total storage for the rows inserted into the TWG.MSMT table is dependent on two items:

- The physical storage required for each row in the table
- The physical storage required for the index defined for each row in the table

These sizes turn out to be straightforward. Each TWG.MSMT row takes 86 bytes, and each row will have a 32-byte index associated with it. The total amount of storage required for the table can be computed as shown in the following box.

TWG.MSMT table sizing: The formula used for TWG.MSMT table sizing is: 118 bytes per TWG.MSMT row * (number of rows inserted per day) * (number of days stored)/1024000000 = GB of required storage

Storage requirements for the TWH_MART database can be based on the same size estimates from the ones for the TWG.MSMT table. Given that the facts moved into the TWG.MSMT table are hourly facts from warehouse pack sources, these facts should move into the TWH_MART hourly fact tables. This leads to a similar size estimate per day for the TWH_MART hourly fact tables as required per day for the TWG.MSMT table.

Additionally, TWH_MART hourly facts are further summarized into daily, weekly, and monthly fact tables. Estimates for storage space requirements for these fact tables can be made from the estimate of hourly fact table storage. Simple formulas for these can be estimated as shown in the following box.

TWH_MART fact table sizing: Here are the formulas that are used in TWH_MART fact table sizing. Note that these formulas are per day and will need to be multiplied by the number of days stored in the TWH_MART database to determine total storage requirements.

- Storage per day for daily facts = storage per day for hourly facts / 24
- Storage per day for weekly facts = storage per day for daily facts / 7
- Storage per day for monthly facts = storage per day for weekly facts / 4

See each product's *Warehouse Pack Guide* for detailed information about setting the pruning time periods for the TWH_MART fact tables.

4.4 A sizing example

To more fully illustrate the concepts from the previous section, here we show a specific example of how to estimate the storage requirements for a Tivoli Data Warehouse V1.3 implementation using the IBM Tivoli Monitoring Base Operating System resource models.

We know that the primary consumer of storage space will be the TWG.MSMT table in the TWH_CDW database and the various fact tables in the TWH_MART database. The other tables in those databases will have some storage requirements. We recommend 20 GB to 30 GB of storage space for these other tables. We calculate this extra space is sufficient for the TWH_CDW database. Allowing a similar amount of extra storage for the TWH_MART database is sufficient as well. For most Tivoli Data Warehouse V1.3 installations, much of this extra storage space will remain unused, but we recommend being over-provisioned rather than under-provisioned.

Now, we start determining how much storage will be required for the measurement facts that will be responsible for the bulk of the storage requirements in the TWH_CDW and TWH_MART databases. First, we describe the customer environment. Then, we describe how we determined the daily number of rows being inserted into the TWG.MSMT table in the TWH_CDW database. Finally, we demonstrate how to convert this value into an actual physical storage size requirement.

A customer is deploying Tivoli Data Warehouse V1.3 in an environment and using IBM Tivoli Monitoring Version 5.1.2 to monitor 5000 endpoints (individual servers) with the base Windows and UNIX operating resource models. The customer has an equal number (2500) of both Windows servers and UNIX servers and plans on keeping 180 days of data in their data warehouse and 90 days of data in the data marts used for reporting. The default resource models that are pushed down to the individual servers using IBM Tivoli Monitoring V5.1.2 are as follows for Windows servers: Windows Processor, Windows Physical Disk, Windows Logical Disk, and Windows Memory. On the UNIX servers, the resource models pushed down are: UNIX CPU, UNIX Memory, UNIX File System, and UNIX Process. The setup and installation are done with IBM Tivoli Monitoring V5.1.2 and do not involve any Tivoli Data Warehouse V1.3 software.

Now, we must determine how all these various resource models generate records that will be inserted into the TWG.MSMT table every day.

Each resource model monitors certain resources on each server. These measurements are summarized at hourly intervals for Tivoli Data Warehouse V1.3. With the resource models selected, we find that each resource has a value for the minimum, maximum, average, and total value during that hour. These

metrics are inserted into a single row in the TWG.MSMT table during the AMX ETL process. The AMX ETL is the ETL that pulls data from the IBM Tivoli Monitoring V5.1.2 database into the TWH_CDW database. The next step is to determine which measurements are recorded per server and which are recorded per resource instance on the server.

To illustrate, we look at the Windows Processor resource model. This resource model measures three metrics per physical server and six additional metrics per physical processor. We referenced this information in the *IBM Tivoli Monitoring Resource Model Reference Guide*, SH19-4570 and verified these facts in our environment.

Because we know there are three metrics per server and six metrics per processor, a simple formula exists to compute how many rows per day will be inserted into the TWG.MSMT table.

Formula: Use the following formula to calculate the number of rows per day inserted into the TWG.MSMT table by the Windows Processor resource model: [(number of physical servers * 3)+(number of physical processors * 6)] * 24 hours

Therefore, for the customer monitoring 2500 Windows servers, we need to also know the number of physical processors *total*. In this case, the customer has a combination of single, dual, and quad processor machines for a total of 4000 processors. So per day, this resource model will be responsible for [(2500*3)+(4000*6)]*24 = 756,000 rows per day into the TWG.MSMT table.

Remember this example, where knowing the number of physical servers being monitored is *not* sufficient information to calculate the number of rows per day generated for a correct sizing estimate. When running the resource models from IBM Tivoli Monitoring V5.1.2, you must understand which metrics are logged per server and which are logged per instance of resource on the server.

We know the customer plans on keeping 180 days worth of data, so we can use the formula from 4.3.3, “Storage size requirements” on page 84 to calculate the total storage requirements for the data generated by this resource model:

118 bytes * 756,000 rows * 180 days/1024000000 bytes per GB = 15.7 GB

Therefore, in addition to the recommended 20 GB to 30 GB for other tables, the customer must plan on having 15.7 GB storage available in the TWH_CDW database for the data from the Windows Processor resource model in the TWG.MSMT table.

We can also estimate the storage requirement of this data in the TWH_MART fact tables using the formulas from 4.3.3, “Storage size requirements” on

page 84. The customer plans on retaining 90 days worth of data in the TWH_MART database. In this case, the customer will need about half as much storage (90 days instead of 180) for hourly fact data in the TWH_MART. This works out to about 7.9 GB of storage for the hourly facts. Using the estimation formulas in 4.3.3, “Storage size requirements” on page 84 for the additional summarized data for the daily, weekly, and monthly fact tables brings the total storage for this data in the TWH_MART database to around 8.3 GB. Remember to allow around 20 GB to 30 GB for the non-fact tables in the TWH_MART database.

This process can be followed to determine the storage requirements for any warehouse pack. Given any particular Tivoli Data Warehouse V1.3 enabled product, first determine the number of rows that will be inserted per day into the TWG.MSMT table. After you have this number, you can easily determine the storage requirements for this table in the TWH_CDW database and make good estimates of the space required for fact tables in the TWH_MART database.

We complete our storage size estimation for this scenario using the spreadsheet tool described in 4.5, “The spreadsheet example tool” on page 89.

For purposes of using the tool, we determine that the resource models will be monitoring the following resources:

- 6000 physical disks total on the Windows servers
- 6000 logical disks total on the Windows servers
- Five processes on each of the 2500 Windows servers with the Windows Memory resource model
- Five mount points on each of the 2500 UNIX servers
- Nine processes on each of the 2500 UNIX servers

Using these values and the spreadsheet tool, we determine that about 5.4 million rows a day will be inserted into the TWG.MSMT table. Storing 180 days of data in the TWH_CDW, the spreadsheet gives us a storage size of about 109 GB storage for the TWG.MSMT table. Storing 90 days of data in the TWH_MART database, the spreadsheet provides an estimate of about 58 GB for the fact tables.

Although this example is based on deploying IBM Tivoli Monitoring V5.1.2 resource models, the process described applies to other product sizings of Tivoli Data Warehouse V1.3. The key point is to understand that the TWG.MSMT table is ultimately the largest consumer of storage in the TWH_CDW. Determining how many rows per day will be inserted into this table is the foundation of any storage sizing project. Armed with that information, you can make reasonable estimates of TWH_MART fact table storage requirements.

4.5 The spreadsheet example tool

We use a simple spreadsheet developed using the previously described process to determine the number of rows per day generated for the TWG.MSMT table for the IBM Tivoli Monitoring V5.1.2 Base Operating System resource models we deploy in our example environment. You can download this spreadsheet using the instructions in Appendix A, “Additional material” on page 271.

You can use it to obtain sizing references for these resource models or as an example of how to build a tool for estimating the storage requirements for the TWG.MSMT table. Remember the size estimates generated by the tool do *not* include the recommended 20 GB to 30 GB of storage for other tables in the TWH_CDW or TWH_MART database.

Important: It is important to note that this spreadsheet is provided on an *as is* basis. Results generated with the spreadsheet are not guaranteed for every type of deployment.

The first page of the spreadsheet helps you calculate the number of rows per day inserted into the TWG.MSMT table. The second page of the spreadsheet enables you to determine the storage requirements for the TWH_CDW TWG.MSMT table and TWH_MART fact tables.

You can use the spreadsheet by changing the values that are highlighted in the yellow boxes to suit your environment. The values calculated from your entries in those boxes are shown in red boxes and should not be changed.

Any changes to the IBM Tivoli Monitoring V5.1.2 resource models will affect the correctness of the values calculated by the spreadsheet. It is advisable to use it as an example of how to determine the storage requirements of Tivoli Data Warehouse V1.3 and not as a total solution for doing so.

4.6 Tivoli Data Warehouse V1.3 sizing summary

We use this chapter to introduce how to generate size requirements for Tivoli Data Warehouse V1.3. If you only take one idea from this information, remember that Tivoli Data Warehouse V1.3 is a DB2 Universal Database Version 8.2 solution and requires database server hardware to run.

We recommend using the DB2 Universal Database Version 8.2 documentation (`http://www.ibm.com/software/data/db2/udb/support/manualsv8.html`) and Redbooks (`http://www.redbooks.ibm.com`) for further reference.

Archived

5

Getting Tivoli Data Warehouse V1.3 up and running

This chapter describes the installation steps for Tivoli Data Warehouse. You can install the application with all the components on a single system or with the components distributed throughout your network.

Always check the *Tivoli Data Warehouse Release Notes*, SC32-1399, for details about required fix packs, patches, or operating system-specific configurations. Look for late-breaking information about the required service at the Tivoli Data Warehouse support Web site, available at:

http://www.ibm.com/software/sysmgmt/products/support/TivoliDataWarehouse.html

Topics covered include:

- Overview of the installation
- Crystal Enterprise installation (optional)
- Installing the control server and creating the databases
- Installing warehouse agents (optional)
- Checking the installation
- Installing warehouse packs

© Copyright IBM Corp. 2005. All rights reserved.

5.1 Overview of the installation

Tivoli Data Warehouse V1.3 runs on top of the data warehouse of IBM DB2 Universal Database Enterprise Edition, and optionally, you can use the included Crystal Enterprise Professional Version 9 for Tivoli for issue reports. All these components (shown in Figure 5-1) can be installed on a single machine, or spread out on several machines. If you choose the single machine installation, you have to use Microsoft Windows, because it is the only platform control server is supported.

In Chapter 3, “Tivoli Data Warehouse design considerations” on page 39, we cover planning and design considerations of Tivoli Data Warehouse V1.3. Based on this, the following points of the system architecture must be clear by now:

- Crystal Enterprise Professional Version 9 for Tivoli can run on Microsoft Windows, AIX, Sun, and Linux (Linux and UNIX versions run under a servlet engine such as WebSphere).
- The control server can be installed on Windows only.
- We can have four data marts and four central data warehouses. Central data warehouses and data marts can be installed on Windows, AIX, Solaris, Linux, and z/OS

When Tivoli Data Warehouse and all of its databases are on a single machine, it is called a quick start deployment. Otherwise, it is the distributed installation.

The next sections show how to install each of these components.

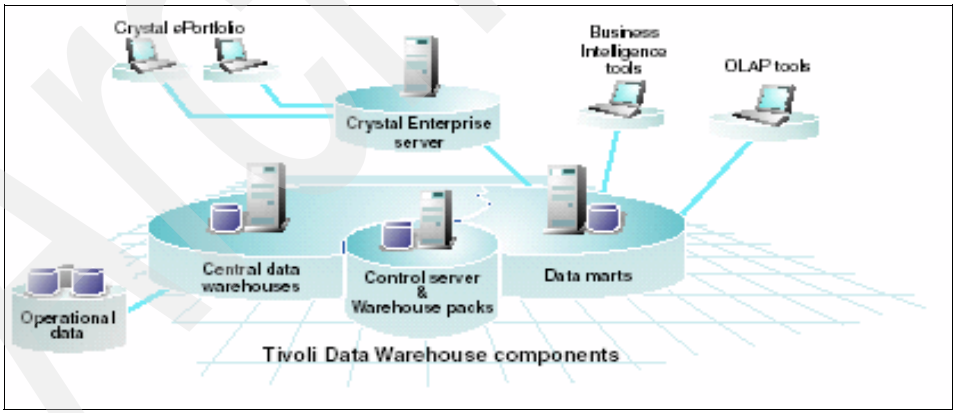


Figure 5-1 Tivoli Data Warehouse components

5.2 Crystal Enterprise installation (optional)

Crystal Enterprise Professional Version 9 for Tivoli is a reporting solution provided with Tivoli Data Warehouse V1.3. Crystal Enterprise typically resides on a system that does not have other Tivoli Data Warehouse components installed on it. Users access Crystal Enterprise reports using a Web browser from any system.

Crystal Enterprise is integrated with Tivoli Data Warehouse V1.3 to provide historical reporting. Warehouse packs provide reports designed for use with Crystal Enterprise. You access the reports with a Web browser that connects to the Crystal Enterprise server.

This task can be accomplished by a Tivoli Specialist or an Operational Support Specialist (no skill on the Tivoli environment is required).

Ensure that the following components are installed and configured correctly before you install Crystal Enterprise Professional for Tivoli. Refer also to 3.1.4, “Crystal Enterprise requirements” on page 44 for additional information.

- Microsoft Windows 2000, Windows 2003 (or AIX, Sun, and Linux all of which require a servlet engine).
- Web server software, such as Microsoft IIS, iPlanet Enterprise Server, or IBM HTTP Server.
- Microsoft Internet Explorer or Netscape Navigator Web browser.
- Check if the server has a fully qualified host name. Issue a **`nslookup machine ip address`** command and check that the command returns a fully qualified host name.
- IBM DB2 Client, in order to have access to the data mart for reporting. Refer to 9.2, “Installing and configuring the DB2 client and server” on page 246 for details.

Note: Because we used a beta version of Tivoli Data Warehouse V1.3 product in this IBM Redbook project, the windows you will see in the general availability version of the product might be slightly different from ours.

Archived

In order to install Crystal Enterprise Professional Version 9 for Tivoli on Windows, complete the following steps.

> **Note:** Use the installation media provided with the Tivoli Data Warehouse V1.3 product. This ensures that you install the correct version of Crystal Enterprise.

1. Run **`setup.exe`** from the win32 directory of your product distribution. On the Welcome window, click **Next** to proceed.
2. Accept the license agreement.
3. As shown in Figure 5-2, change the destination folder (optional) and choose the installation type **New - Install a new Crystal Enterprise System**. Click **Next**.

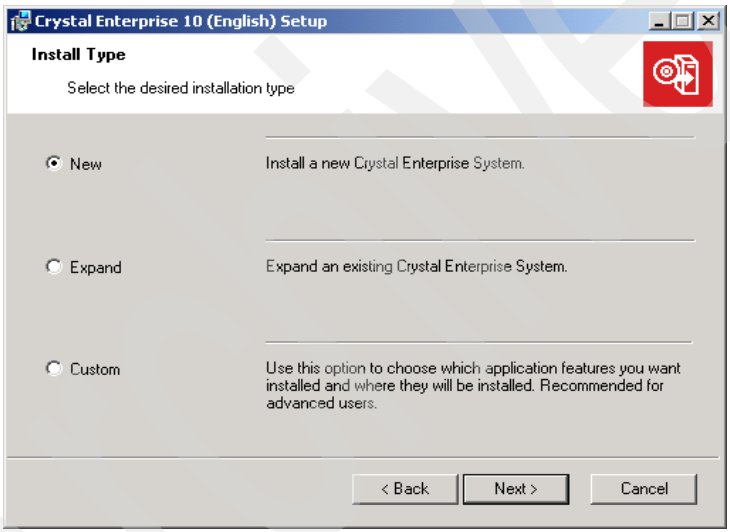


Figure 5-2 Installation type

4. The setup program checks to see whether or not the Microsoft Data Engine (MSDE) or Microsoft SQL server is installed on the local machine. If the setup program detects a database, use the Microsoft SQL Server Authentication window to provide the credentials that correspond to the database account you set up for the APS. The default user ID for the database account is named `sa`. For more information, refer to 3.1.4, “Crystal Enterprise requirements” on page 44.

   If the setup program does not detect an existing database, the installation process will install Microsoft Data Engine and will create the credentials for the default SQL administrator account (`sa`) user ID. The installation wizard

prompts you for the password to be used by the `sa` user ID. The setup program later configures the APS to connect to its system database using the `sa` account and the password you create here. Click **Next**.

Figure 5-3 shows the Crystal Enterprise Professional Version 9 for Tivoli installation in progress.

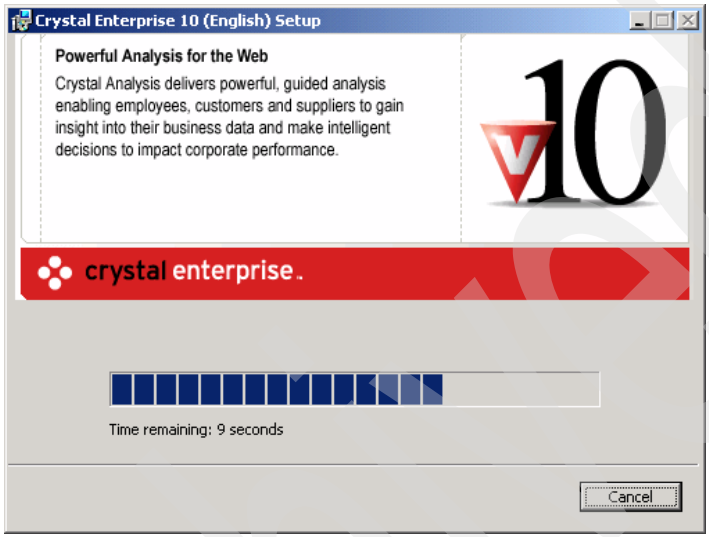


Figure 5-3 Installation window

Archived

5. Click **Finish** on the completion window, as shown in Figure 5-4.

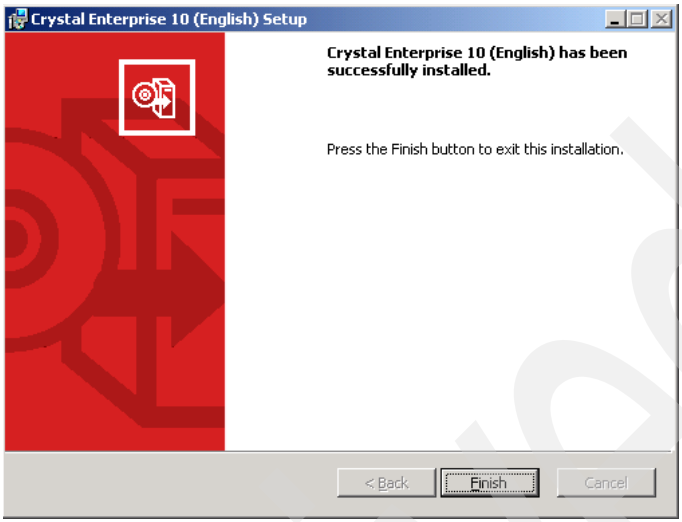


Figure 5-4 Completion window

6. If the Web server installed on the local machine is a supported version, the setup program installs and configures the appropriate Crystal Enterprise Web Connector. Therefore, when the installation is complete, you can access the Crystal Enterprise Professional for Tivoli server by opening your Web browser and pointing it to:

   `http://<CRYSTALSERVER>/crystal/enterprise10/`

   In this URL, `<CRYSTALSERVER>` is the host name of the Crystal Enterprise Professional for Tivoli server machine.

Archived

7. Click the **Admin Launchpad** link. This launches the Admin Launchpad. See Figure 5-5.

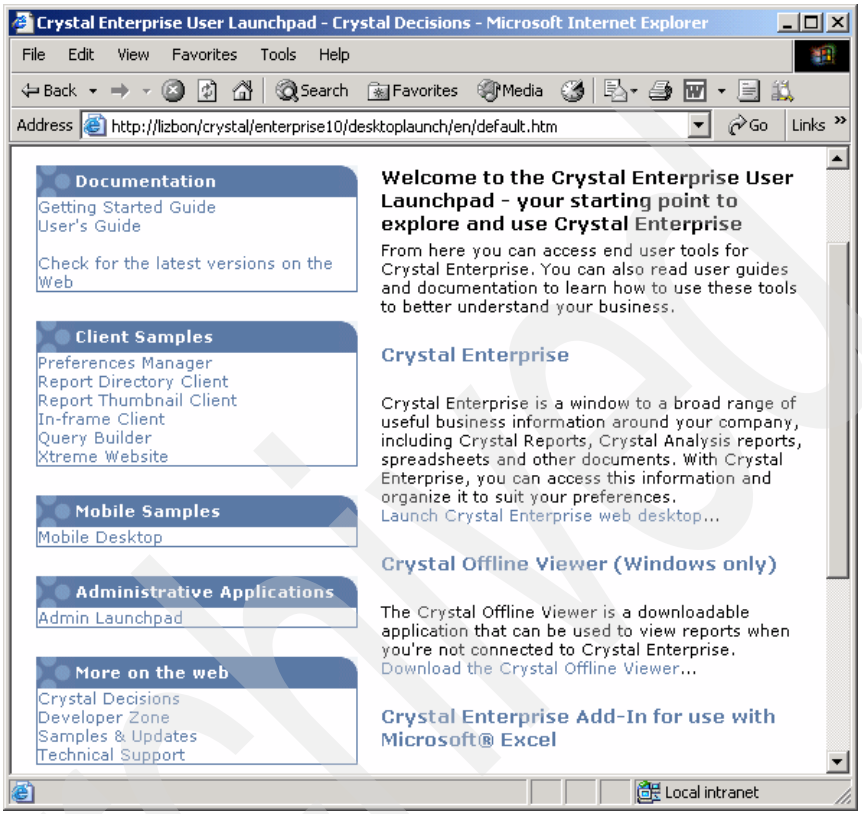


Figure 5-5 Crystal Enterprise Launchpad window

Archived

8. Crystal Enterprise Launchpad is launched, as shown in Figure 5-6. Click **Administrative Tool Console**.

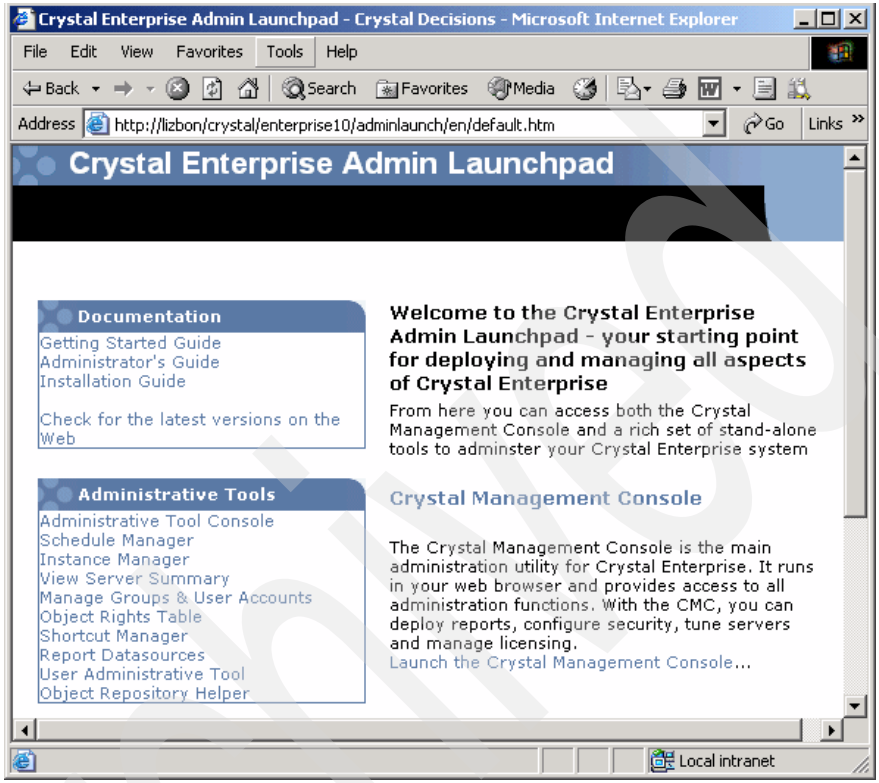


Figure 5-6 Crystal Enterprise Admin Launchpad

Archived

9. Log on as `Administrator`, as shown in Figure 5-7, to verify that the setup program installed and configured the appropriate Crystal Enterprise Web Connector. The default password for the Administrator account is set to blank (no password).

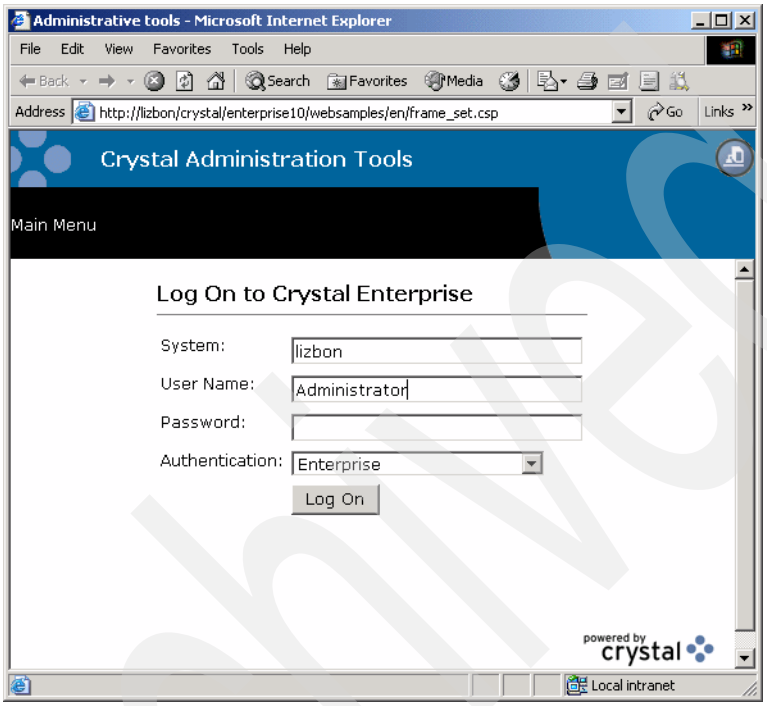


Figure 5-7 Crystal Administration Tools

For details about how to use the Crystal Enterprise Professional for Tivoli Web interface, refer to the *Crystal Enterprise Professional Version 9 for Tivoli Administrator's Guide*, which is shipped with Crystal Enterprise Professional Version 9 for Tivoli.

Post-installation steps

In order to work with Tivoli Data Warehouse V1.3, it is necessary to set up the IP port where the Crystal Report Application Server will be listening. The suggested port is the 1566, but you can use any available port. You need this information when you install the central server of the Tivoli Data Warehouse V1.3. Use the following steps to set up the Crystal Report Application Server:

1. On the Windows task bar, click **Start** → **Programs** → **Crystal Enterprise 10** → **Crystal Configuration Manager**.

Archived

2. Highlight **Crystal Report Application Server**, as shown in Figure 5-8. Right-click and select **Stop** to stop the server.

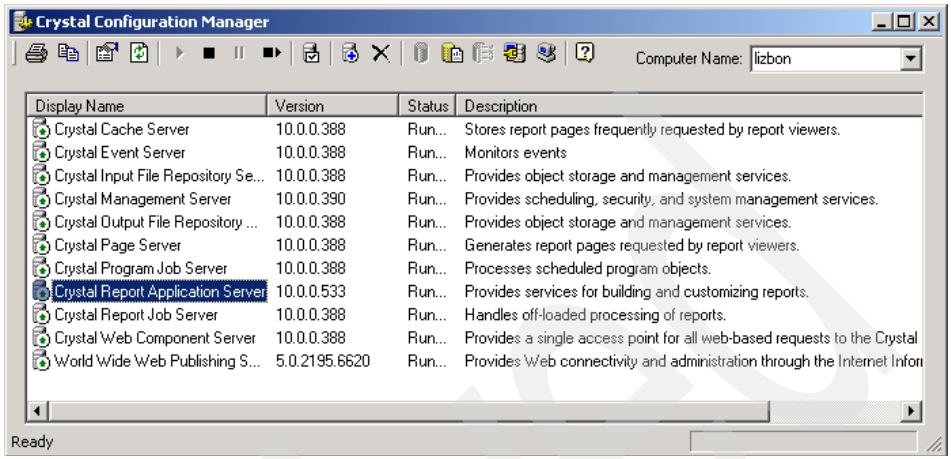


Figure 5-8 Crystal Configuration Manager window

3. Right-click **Properties**. In the Properties tab, edit the Command text box by adding the option `-ipport 1566` at the end. See Figure 5-9 on page 101.

Archived

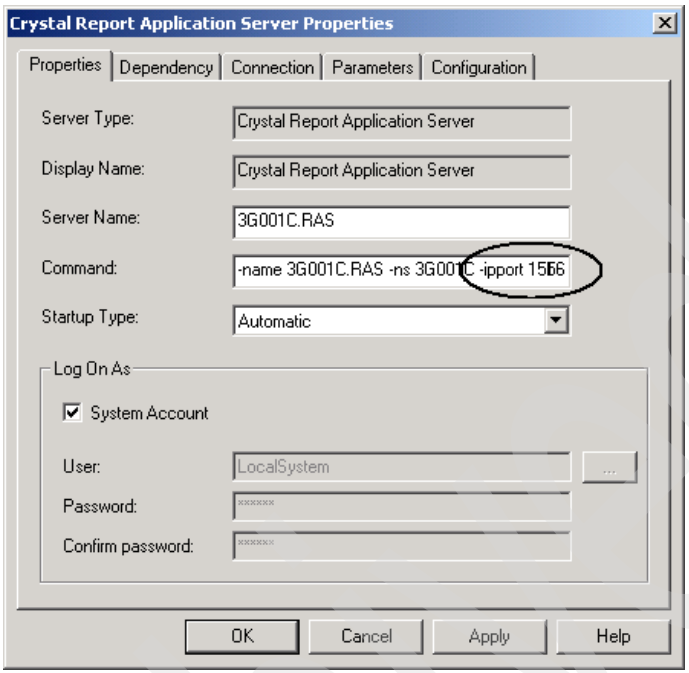


Figure 5-9 Changing the Report Application Server port

4. Click **OK** to close the window. Right-click again in **Crystal Report Application Server** and choose **Start**. Close the Crystal Configuration Manager window.

5.3 Installing the control server and creating the databases

The control server is the most important component of Tivoli Data Warehouse. It is supported on Windows only. We cover the installation in this section. The installation can be accomplished by a Tivoli Specialist.

Tip: For the control server installation and creating the databases, you do not need to have DB2 administration skills. After you get the correct database information (such as host name, port number, and so on) from your DBA, you can perform the installation. But for administrating and maintaining a Tivoli Data Warehouse environment, DB2 administration skills are required. Refer to Chapter 9, “IBM DB2 UDB administration for Tivoli Data Warehouse” on page 243 for a discussion about DB2 administration.

Ensure that the following components are installed and configured correctly before you install Crystal Enterprise Professional for Tivoli. Refer also to 3.1.4, “Crystal Enterprise requirements” on page 44 for additional information.

- Check all the prerequisites.
- Check if the server has a fully qualified host name. Issue a **`nslookup machine name`** command and check that the command returns a fully qualified host name.
- If you intend to use Crystal Enterprise Professional for Tivoli, make sure you have already installed it.
- Install or upgrade, or both, IBM DB2 Universal Database Enterprise Edition to Version 8.2. Note that DB2 V8.2 is also known as DB2 V8.1 Fix Pack 7. If you are not a DB2 DBA and intend to install the DB2 by yourself, see Chapter 9, “IBM DB2 UDB administration for Tivoli Data Warehouse” on page 243. For a distributed installation, it is necessary to install the IBM DB2 Universal Database Enterprise Edition on all servers that will hold the data marts, central data warehouses, and the control server. On the report server, you must install the DB2 run-time client.

5.3.1 Quick deployment

In the quick deployment, the control server and the databases are created in one machine. This machine should be a Windows machine. This configuration is more useful in small and test environments.

In order to perform the Tivoli Data Warehouse V1.3 installation on Windows, use the following steps:

1. Insert the Tivoli Data Warehouse V1.3 installation CD into the CD-ROM drive. If the installation wizard does not start up, run **`setup.exe`**, which is located in the Windows directory of the CD.
2. Select the language you want the wizard to use, and click **OK**.
3. In the Welcome window, click **Next**.
4. The Tivoli Common Logging Directory window displays the name of the Tivoli common logging directory. If Tivoli Data Warehouse is the first Tivoli software product on your system that uses this directory, specify a location, and then click **Next**.
5. In the Destination Directory window, specify the installation directory for Tivoli Data Warehouse. The default directory is %ProgramFiles%\TWH\ on Windows and /opt/twh/ on UNIX. Then click **Next**.

Tip: Choose a destination path without spaces in it; otherwise, you might have problem running some tools after the installation.

6. In the Installation Type window, select **Quick start**, and then click **Next** (Figure 5-10).

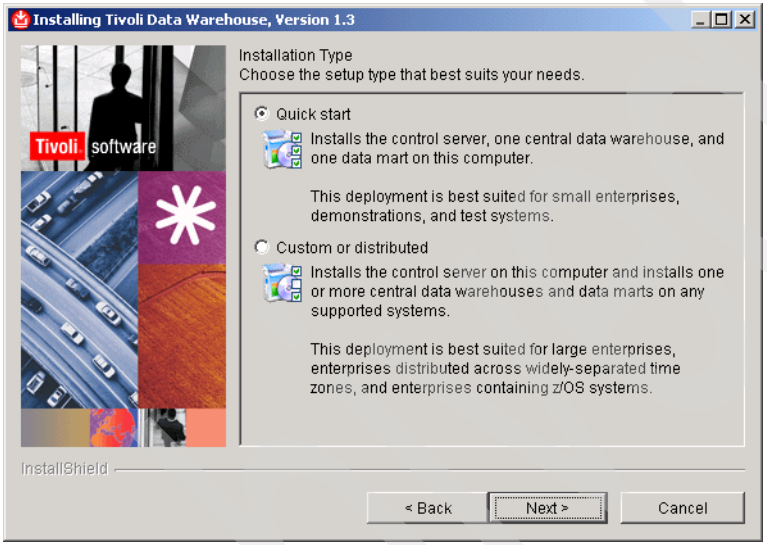


Figure 5-10 Installation Type window

7. In the Local DB2 Connection window, specify the existing DB2 user ID and password to use when creating and connecting to Tivoli Data Warehouse databases on this computer, and then click **Next**.
8. If you do not have a Crystal Enterprise server or do not use Crystal Enterprise for your reporting, in the Host Name field, type `NO_CRYSTAL`. You need to enter a value in the User Name field, but the value is irrelevant and is ignored. Go to step 10 on page 104.
9. If Crystal Enterprise is your choice of reporting tools, in the Crystal Enterprise Connection window, specify the following connection information for the Crystal Enterprise server, and then click **Next**:
   - For Host Name, type the fully qualified host name of the Crystal Enterprise server.
   - For User Name, type your Crystal user name. The default user name for the Crystal Enterprise server is Administrator.
   - For Password, type your Crystal password. By default, there is no password.

– For Crystal Management Server Port Number, type the port number for Crystal Management Server. The default port number is 6400.

– For Report Application Server Port Number, type the port number for Report Application Server. The default port number is 1556.

Important: Make sure that you have executed the post-installation step in 5.2, “Crystal Enterprise installation (optional)” on page 93.

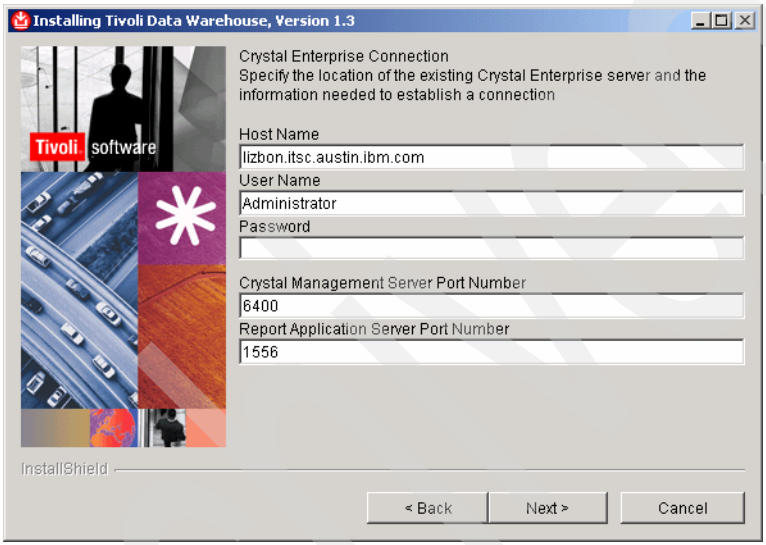


Figure 5-11 Crystal Enterprise Connection window

10. In the Language to Install window, select the language, and then click **Next**.

11. The Summary window (Figure 5-12 on page 105) indicates that you are installing the control server, one central data warehouse, and one data mart on the local computer. Click **Install** to start the installation.

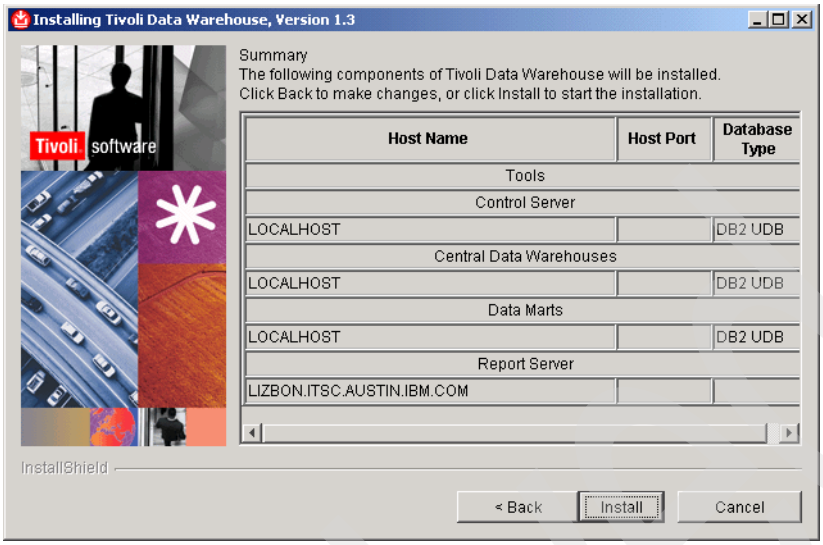


Figure 5-12 Summary window

12. After the installation is over, a completion window opens. It has a successful completion notice or messages describing problems. Make sure the window does not list any warnings or errors. Click **Next**. If you are prompted to restart, click **Yes, restart my system**.

5.3.2 Custom or distributed deployment

In this deployment, you are able to define where you want to put the data mart and central data warehouse databases. Also, you can create these database before the installation. Therefore, your DBA will be able to create databases that better fit your environment. This deployment is better suited for big and mid-sized production environments. See Chapter 7, “Performance maximization techniques” on page 175 for more information about this.

In order to perform the Tivoli Data Warehouse V1.3 installation on a Windows workstation, use the following steps:

1. Insert the Tivoli Data Warehouse V1.3 installation CD into the CD-ROM drive. If the installation wizard does not start up, run **`setup.exe`**, which is located in the Windows directory of the CD.
2. Select the language you want the wizard to use, and click **OK**.
3. In the Welcome window, click **Next**.

4. The Tivoli Common Logging Directory window displays the name of the Tivoli common logging directory. If Tivoli Data Warehouse is the first Tivoli software product on your system that uses this directory, specify a location, and then click **Next**.
5. In the Installation Type window, select **Custom or distributed**, and then click **Next** (Figure 5-13).

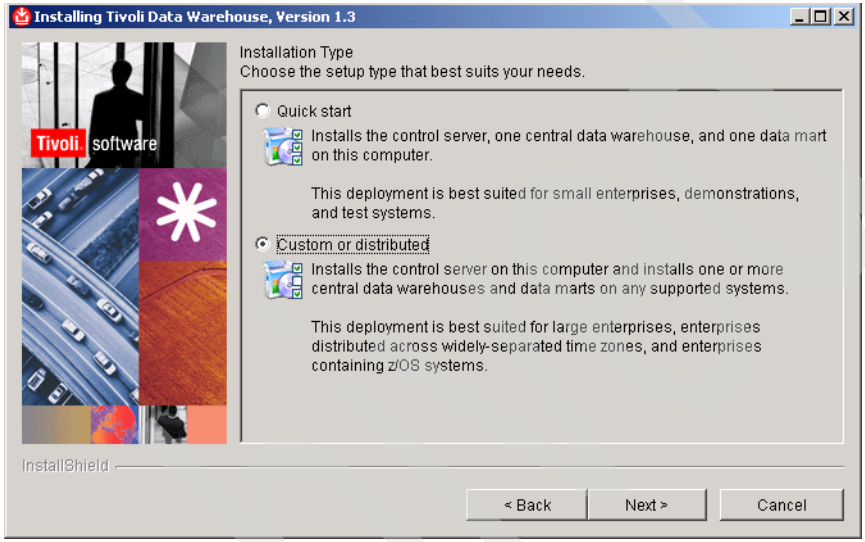


Figure 5-13 Install Type window: Custom installation

6. In the Local DB2 Connection window, specify the existing DB2 user ID and password to use when creating and connecting to Tivoli Data Warehouse databases on this computer, and then click **Next**.
7. In the Central Data Warehouses window, specify the computers on which to create a central data warehouse database. Click **Add**.
8. In the Central Data Warehouse Properties window, enter the following values (Figure 5-14 on page 107):
   - For Host Location, choose Local or Remote, depending on the location of the central data warehouse database. Select Remote if you would like to install on a different computer then the control server.
   - For Host Name, type the fully qualified host name of the computer.
   - For DB2 Port Number, type the port number for DB2 (the default port for DB2 is 50000). If you are not sure, ask your DBA.
   - For DB2 User ID and Password, type the user ID and password that is used to connect to the central data warehouse on this system.

– For Database Type, select the type of database. If you do not know what value to enter, ask your DBA.

– Select the **Use Preconfigured Database** option if you or your DBA have preconfigured a central data warehouse database to use in the Tivoli Data Warehouse environment.

– For Preconfigured Database Name, type the name of the preconfigured database.

Click **Next**.

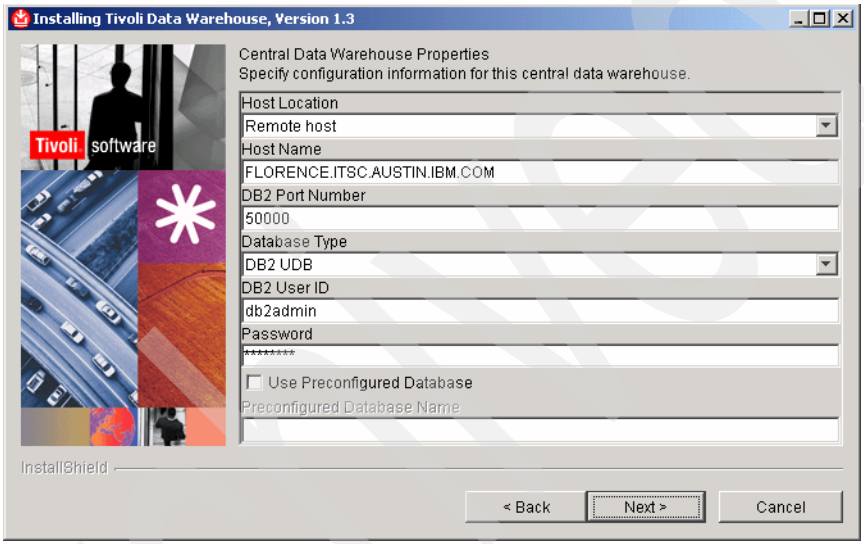


Figure 5-14 Central Data Warehouse Properties window

9. The table in the Central Data Warehouses window now lists the central data warehouse you just configured. Repeat step 8 on page 106 to define each central data warehouse in your deployment. When the list is complete (Figure 5-15 on page 108), click **Next**.

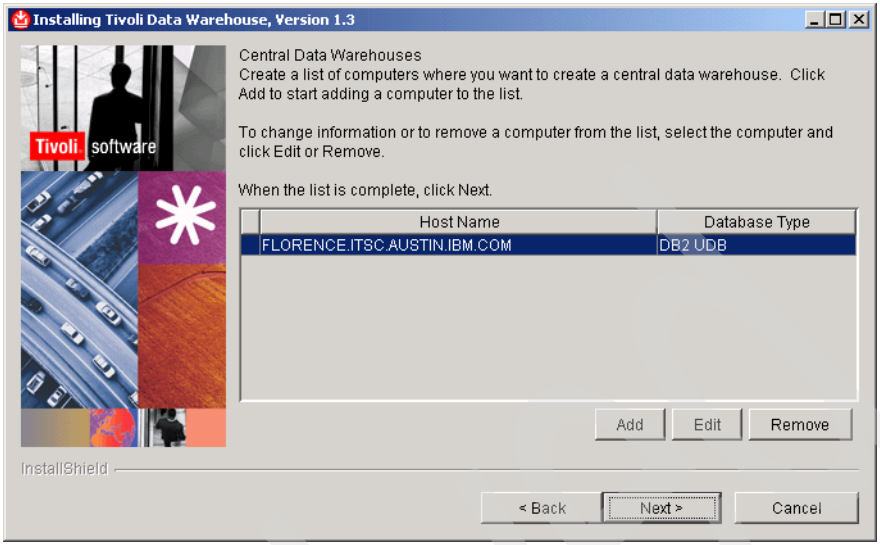


Figure 5-15 Central Data Warehouse information: Completed

10. In the Data Marts window, specify the computers on which to create a data data mart databases. Click **Add**.
11. In the Data Mart Properties window, enter the following values (Figure 5-16 on page 109):
    - For Host Location, choose Local or Remote, depending on where the data mart database will be.
    - For Host Name, type the fully qualified host name of the computer.
    - For DB2 Port Number, type the port number for the DB2 (the default port for DB2 is 50000). If you do not know, ask your DBA.
    - For DB2 User ID and Password, type the user ID and password that is used to connect to the central data warehouse on this system.
    - For Database Type, select the type of database. If you do not know, ask your DBA.
    - Select the **Use Preconfigured Database** option if you or your DBA have preconfigured a central data warehouse database to use in the Tivoli Data Warehouse environment.
    - For Preconfigured Database Name, type the name of the preconfigured database.

    Click **Next**.

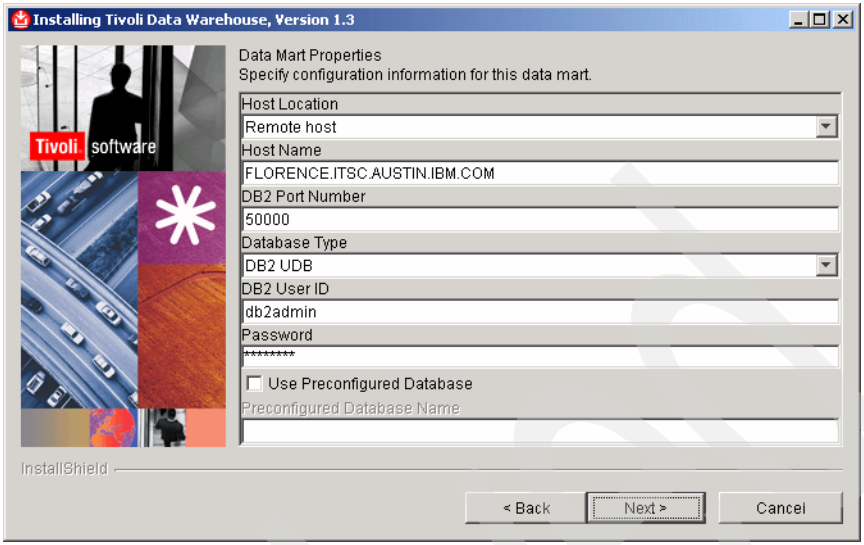


Figure 5-16 Data Mart Properties window

12. The table in the Data Marts window now lists the central data warehouse you just configured (Figure 5-17). Repeat step 9 to define each central data warehouse in your deployment. When the list is complete, click **Next**.

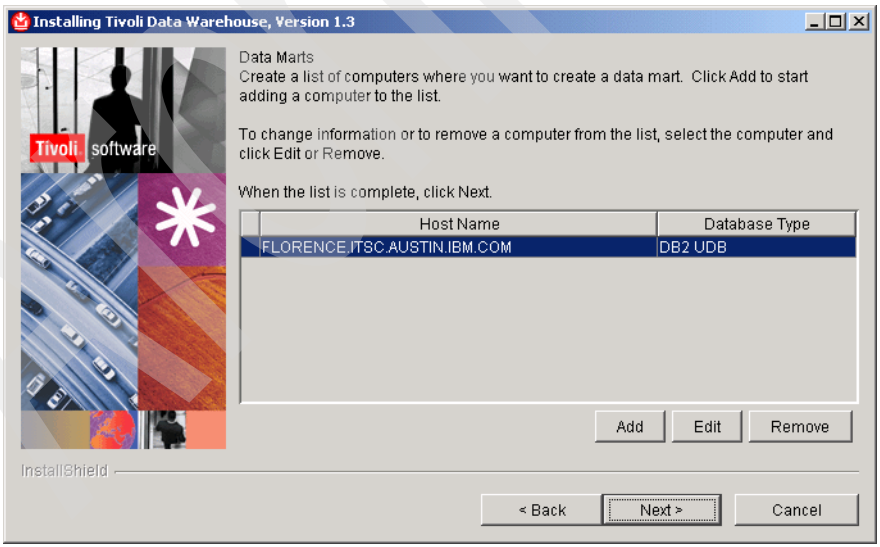


Figure 5-17 Data Mart information: Completed

13. If you do not have a Crystal Enterprise server or do not use Crystal Enterprise for your reporting, in the Host Name field, type `NO_CRYSTAL`. You need to enter a value in the User Name field, but the value is irrelevant and is ignored. Skip to step 15 on page 111.

14. If Crystal Enterprise is your choice of reporting tools, in the Crystal Enterprise Connection window, specify the following connection information for the Crystal Enterprise server, and then click **Next** (see Figure 5-18):
    - For Host Name, type the fully qualified host name of the Crystal Enterprise server.
    - For User Name, type your Crystal user name. The default user name for the Crystal Enterprise server is Administrator.
    - For Password, type your Crystal password. By default, there is no password.
    - For Crystal Management Server Port Number, type the port number for Crystal Management Server. The default port number is 6400.
    - For Report Application Server Port Number, type the port number for Report Application Server. The default port number is 1556.

> **Important:** Make sure you have executed the post-installation step in 5.2, “Crystal Enterprise installation (optional)” on page 93.

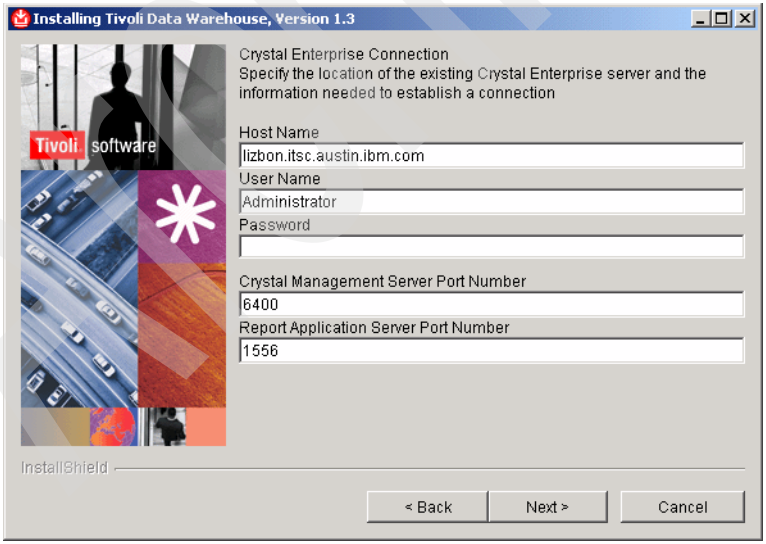


Figure 5-18 Crystal Enterprise Connection window

15. In the Language to Install window, select the language, and then click **Next**.
16. The Summary window (Figure 5-19) indicates that you are installing the control server on the local computer and one central data warehouse and one data mart on the remote computer. Click **Install** to start the installation.

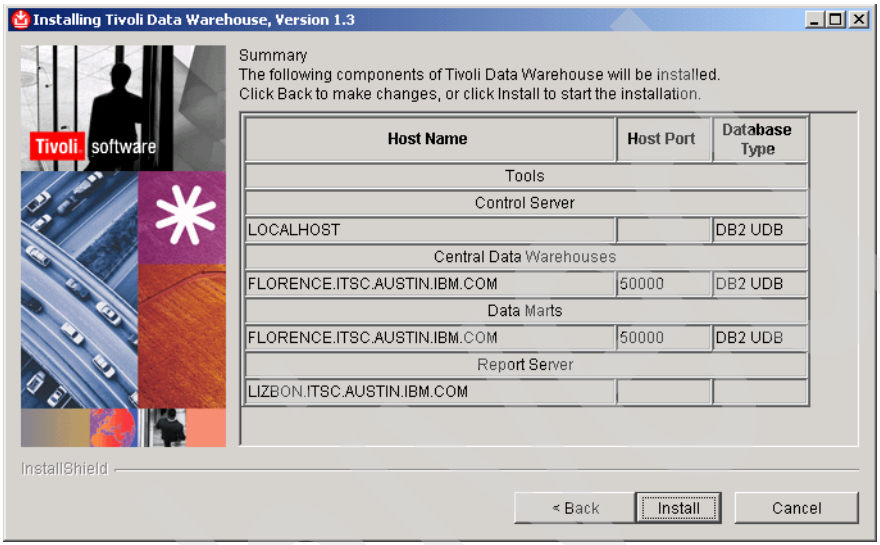


Figure 5-19 Summary window

17. After the installation is over, a completion window opens. It has a successful completion notice or messages describing problems. Make sure that the window does not list any warnings or errors. Click **Next**. If you are prompted to restart, click **Yes, restart my system**.

5.3.3 Finishing the installation

This section contains important post-installations instructions for configuring the environment.

Configuring IBM DB2 Warehouse Control Database Management

In order to configure IBM DB2 Warehouse Control Database Management, complete the following steps:

1. Start Warehouse Control Database Management. Open the Data Warehouse Center - Control Database Management window by selecting **Start** → **Programs** → **IBM DB2** → **Setup-tools** → **Warehouse Control Database Management**.

2. Type `TWH_MD` in the new control database.
3. Do not change the schema name.
4. Type the IBM DB2 instance owner user ID and password for the control database, and then click **OK**.
5. You will see a Progress window, as shown in Figure 5-20. When the "`Processing has completed`" message appears, click **Close**.

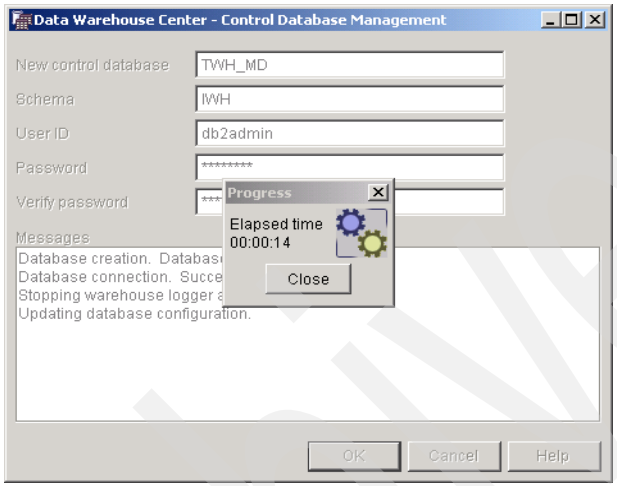


Figure 5-20 Configuring DB2 Warehouse Control Database Management

Creating ODBC connections to the data mart databases

The Crystal Enterprise server needs to have access to the information stored in the data mart database for reporting. After the installation of Tivoli Data Warehouse V1.3 is finished, one data mart database is created (TWH_MART) and an ODBC connection for this database must be set up on the Crystal Enterprise server.

Tivoli Data Warehouse V1.3 provides the twh_create_datasource script that sets up the ODBC connections to the data mart databases. You can use this script or create the ODBC connections manually. In order to create an ODBC connection to the TWH_MART database using the twh_create_datasources script, perform the following tasks:

1. On the Crystal Enterprise server machine, open an IBM DB2 command window by selecting **Start** → **Programs** → **IBM DB2** → **Command Window**.

2. Run the **twh_create_datasource** script using the following syntax, as shown in Example 5-1:

```
twh_create_datasource <DBtype> <ID> <odbcname> <DBname> <SRVname> <port>
```

Where:

<DBtype>	Can be set to DB2UDB or DB390 depending on the location of the database.
<ID>	A unique identifier for the local node name. The script creates node names following the naming convention: TDWCS%ID%. In our example, we used ID=10, resulting in node name TDWCS10.
<odbcname>	The ODBC data source name.
<DBname>	The data mart database name.
<SRVname>	The IBM DB2 server in which the data mart database resides. This must be the fully qualified host name.
<390LocalDBName>	For databases on z/OS only. Specifies the local database name.
<port>	The port number to connect to the IBM DB2 server.

Example 5-1 The twh_create_datasource script

```
C:\Temp>twh_create_datasource.bat DB2UDB 10 TWH_MART TWH_MART
florence.itsc.austin.ibm.com 50000

Creating DB2/UDB datasource TWH_MART

C:\Temp>db2cmd /w /c /i db2 catalog tcpip node TDWCS10 remote
florence.itsc.austin.ibm.com server 50000
DB20000I  The CATALOG TCPIP NODE command completed successfully.
DB21056W  Directory changes may not be effective until the directory cache is
refreshed.

C:\Temp>db2cmd /w /c /i db2 catalog database TWH_MART at node TDWCS10
authentication server
DB20000I  The CATALOG DATABASE command completed successfully.
DB21056W  Directory changes may not be effective until the directory cache is
refreshed.

C:\Temp>C:\Temp\ODBCcfg.exe DB2 TWH_MART TWH_MART

No Username was provided. Skipping connection test.

C:\Temp>
```

5.4 Installing warehouse agents (optional)

If you have installed one data mart or central data warehouse in a different computer than the control server, you can install warehouse agents in these computers.

The warehouse agent is the component of IBM DB2 Warehouse Manager that manages the flow of data between data sources and target databases that are on different computers. During the installation of the Tivoli Data Warehouse, a local warehouse agent on the control server is installed to manage the data flow between operational data sources, central data warehouse, and data mart databases.

Typically, you place an agent on the computer that is the target of a data transfer. From the Tivoli Data Warehouse perspective, that computer will become a *remote warehouse agent site*, or simply an *agent site*, after registering and enabling the warehouse agent to run ETLs.

IBM DB2 Data Warehouse Center will then use the remote warehouse agent site machine to manage the transfer of Tivoli Data Warehouse data. This can speed up the data transfer and reduce the workload on the control server. The computer on which you install a warehouse agent is called an agent site.

Note: Make sure that the warehouse agent site machines that will run ETL1 can connect to the operational data source databases.

Make sure that the warehouse agent site machines that will run ETL2 can connect to the corresponding central data warehouse databases and data mart databases.

There are two steps to be performed in order to create remote warehouse agent sites:

1. Install IBM DB2 Warehouse Manager on every server that will become warehouse agent sites. See Chapter 9, “IBM DB2 UDB administration for Tivoli Data Warehouse” on page 243.
2. On the machine that will become the warehouse agent site, use the Tivoli Data Warehouse installation wizard to register and enable the warehouse agent to run ETLs.

After the warehouse agents have been registered with the control server, the following steps should be performed:

1. On the remote agents site, catalog the databases that the remote agent is supposed to use.

2. On the remote agents site, make available the warehouse pack files.
3. On the data warehouse control servers site, configure the ETL processes to use the remote agent.

Warehouse agents are supported on Windows and UNIX systems only. If you are using IBM DB2 databases on a z/OS system, you must use the warehouse agent on another computer in your deployment.

Creating the remote agent sites

Perform the following steps on each computer that will be a remote agent site. In order to create a remote agent site, IBM DB2 Warehouse Manager must have been installed on each computer. This procedure will register existing IBM DB2 warehouse agents on the Tivoli Data Warehouse control server and will enable the warehouse agent to run ETL processes.

> **Note:** Do not perform this step on the Tivoli Data Warehouse control server. It should be performed on each computer that will become a remote agent site.

To create remote agent sites on a Windows or UNIX system, perform the following steps:

1. Insert the Tivoli Data Warehouse V1.3 CD into your CD-ROM drive.
2. Start the Tivoli Data Warehouse installation wizard using the command for your operating system:

   On Windows: If the Tivoli Data Warehouse installation wizard does not start automatically, run the **ra_setup.exe** program, which is located in the root directory of the CD.

   On UNIX: Run the **ra_setup_unix.sh** program.
3. Select the language in which to view the installation wizard, and then click **Next**.
4. In the Welcome window, click **Next**.
5. The Tivoli Common Logging Directory window displays the name of the Tivoli common logging directory. Click **Next**.
6. In the Local DB2 Connection window, type the user ID and password to use to connect to DB2 Universal Database on this computer, and then click **Next**.
7. In the Destination Directory window, select the location for the Tivoli Data Warehouse files, and then click **Next**.

Archived

8. In the Connection to Remote control server window, specify the following information, and then click **Next**:
   - The fully qualified host name of the existing control server on another computer
   - The port number for DB2 Universal Database on the control server
   - The user ID and password to use to connect to DB2 Universal Database on the control server
9. The Summary window, as shown in Figure 5-21, indicates that you are creating an agent site. Click **Install** to start the installation.

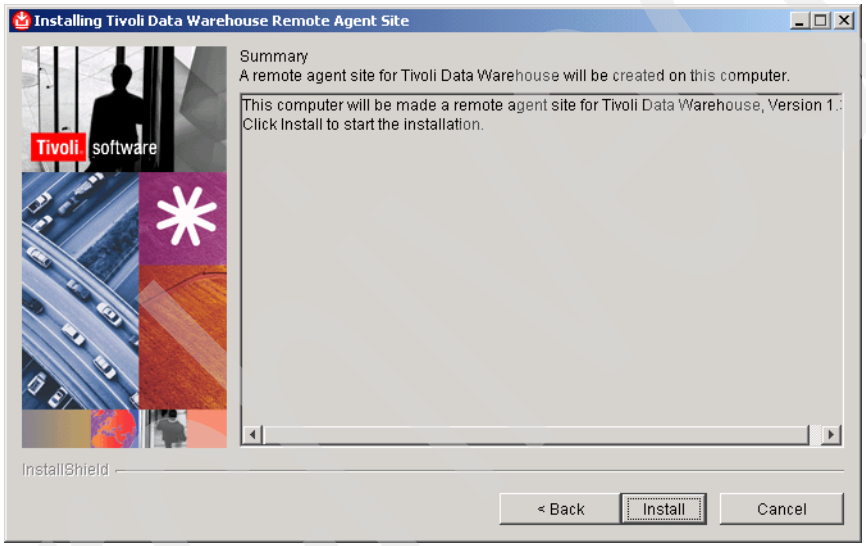


Figure 5-21 Summary window

10. In the Progress window, review the progress of the program. When the program completes, the Installation Results window contains either a successful completion notice or messages describing problems. Make sure that the window does not list any warnings or errors, and then click **Next**.
11. Click **Finish** to exit. You must log out and then log back on to the system before the agent site can be used by any warehouse packs.

5.5 Checking the installation

In this section, we describe the steps to verify the installation of various Tivoli Data Warehouse products.

Note: To check the installation, we use some tools described in the following sections. All these tools are under the directory tools\bin under the root Tivoli installation directory. The tools in the AIX environment have the same name, except the suffix. When using AIX, you must exchange .bat with .sh.

Verify Tivoli Data Warehouse control server

In order to verify the Tivoli Data Warehouse control server, issue the command **twh_list_cs.bat**. If the control server was installed successfully, this command gives you the host name where the control server was installed and the name of the control server database. However, the control server database must be on the same host as the control server.

Example 5-2 shows the output of the **twh_list_cs.bat** command for our case study installation. You will find host name and database name in Example 5-2, which provides an overview of our case study scenario.

Example 5-2 Verify control server (twh_list_cs)

```
C:\TWH\tools\bin>twh_list_cs.bat
Listing the control server information in the Tivoli Data Warehouse registry.

Control Server:
Control Server Database Server Information:
     Host name: florence.itsc.austin.ibm.com
     Vendor: DB2 UDB
     Port: 50000
     Database name: TWH_MD
  Control Server Database Client Information:
     Node name: Not applicable.
     Database alias: TWH_MD
     ODBC connection name:  TWH_MD
  Tivoli Data Warehouse component version: 1.3.0.0
```

Verify Tivoli Data Warehouse central data warehouse databases

Use the batch **twh_list_cdws.bat** command to display information about the central data warehouse databases. Example 5-3 on page 118 shows the output for our case study installation. Both central data warehouses are displayed. The second installed central data warehouse on z/OS has TCDW1 as the database alias and TWH_CDW1 as the ODBC connection name assigned.

Archived

Example 5-3 Verify central data warehouse (twh_list_cdws)

```
C:\TWH\tools\bin>twh_list_cdws.bat
Listing the central data warehouse information in the Tivoli Data Warehouse
registry.

Central Data Warehouse:
  Central Data Warehouse Database Server Information:
     Host name: florence.itsc.austin.ibm.com
     Vendor: DB2 UDB
     Port: 50000
     Database name: TWH_CDW
  Control Server Database Client Information:
     Node name: Not applicable.
     Database alias: TWH_CDW
     ODBC connection name:  TWH_CDW
  Tivoli Data Warehouse component version: 1.3.0.0
  Enabled/Disabled:  E
```

Verify Tivoli Data Warehouse data mart databases

To check the data warehouse data mart databases, use the **twh_list_marts.bat** command. Example 5-4 shows the output for our case study scenario. You can notice the difference between DB2 UDB for OS/390 and z/OS databases and IBM DB2 Universal Database Enterprise Edition DB2 databases. However, you see no differences between Windows-based and UNIX-based databases. In our case study, the TWH_MART database resides on an AIX box.

Example 5-4 Verify data mart databases

```
C:\TWH\tools\bin>twh_list_marts.bat
Listing the data mart information in the Tivoli Data Warehouse registry.


Data Mart:
  Data Mart Database Server Information:
     Host name: florence.itsc.austin.ibm.com
     Vendor: DB2 UDB
     Port: 50000
     Database name: TWH_MART
  Control Server Database Client Information:
     Node name: Not applicable.
     Database alias: TWH_MART
     ODBC connection name:  TWH_MART
  Tivoli Data Warehouse component version: 1.3.0.0
  Enabled/Disabled:  E
```

Archived

Verify remote agents sites

To verify the remote agent sites, from the Windows desktop, select **Start** → **Programs** → **IBM DB2** → **Control Center** to open the DB2 Control Center. From the DB2 Control Center, select **Tools** → **Data Warehouse Center** to open the Data Warehouse Center. In the Data Warehouse Center, select **Administration** → **Agent Sites**.

In this window, all remote agents will be listed. However, in addition to the agents listed by the **twh_list_agentsites** command, the agents for internal use will also be listed.

Verify the installation of the Crystal Enterprise Professional for Tivoli server (twh_update_rptsrv)

To verify the installation of the Crystal Enterprise Professional for Tivoli server and its registration with the Tivoli Data Warehouse control server, enter the command **twh_update_rptsvr -l** from the DOS shell. Example 5-5 shows the output of this command for the case study installation.

Example 5-5 Verify Crystal Enterprise Professional for Tivoli installation

```
C:\TWH\tools\bin>twh_update_rptsrv -l
Report Server Host Name: lizbon.itsc.austin.ibm.com
CMSPORT Crystal Management Server Port Number: 6400
RASPORT Report Application Server Port Number: 1556
Report Server User ID: Administrator
Press any key to continue . . .
```

Verify users, sources, and targets

Enter the **twh_update_userinfo -l** command to get an overview of the used user names and the available sources and targets, as shown in Example 5-6.

Example 5-6 Verify database user

```
C:\TWH\tools\bin>twh_update_userinfo -l


florence.itsc.austin.ibm.com:
    50000:
        TWH_CDW:
            db2admin:
                AMX_TWH_CDW_Source
                AMX_TWH_CDW_Target
                AMY_TWH_CDW_Source
                CDW_TWH_CDW_Source
                CDW_TWH_CDW_Target
        TWH_MART:
```

```
                db2admin:
                     AMY_TWH_MART_Target
                     CDW_TWH_MART_Source
                     CDW_TWH_MART_Target
           TWH_MD:
                db2admin:
                     CDW_TWH_MD_Source

CDWTD0002I The command that changes user IDs and passwords
(twh_update_userinfo)
 completed.
Press any key to continue . . .
```

5.6 Installing warehouse packs

A warehouse pack is the part of a Tivoli software product that provides warehouse functionality. It can be provided on the installation media for the product, on a separate CD, or in a collection of warehouse packs. When provided on the installation media for the product, a warehouse pack is located in a subdirectory named tedw_apps_etl.

Chapter 6, “IBM Tivoli NetView warehouse pack” on page 123 provides instructions about how to install and configure a sample warehouse pack.

However, before installing any warehouse pack, we recommend that you complete the following tasks:

1. Review the implementation guide manual for each warehouse pack you plan to install. The implementation guide manual is located on the installation media of the warehouse pack. For warehouse packs designed for Tivoli Enterprise Data Warehouse V1.1, it is usually in the tedw_apps_etl/<productcode>/pkg/version/doc subdirectory, and for warehouse packs designed for Tivoli Data Warehouse V1.3, it is typically in the <productcode>/doc subdirectory of the warehouse pack installation media, where <productcode> is the AVA code of the product.
2. Define and create warehouse agent sites. See 5.4, “Installing warehouse agents (optional)” on page 114.
3. If the warehouse pack provides an ETL1, which reads data from operation data sources, make sure that the warehouse agent site that will run the ELT1 can connect to the operational data source databases.
4. Back up your Tivoli Data Warehouse deployment to ensure that you can return to a known valid state if you encounter an unrecoverable error.

5. Check if any other ETL process steps for any other warehouse packs that are already installed are scheduled to run during the warehouse pack installation. Change the mode of the existing ETL steps to **Test** to prevent them from running.
6. Verify the warehouse pack installation by issuing the following command on the Tivoli Data Warehouse control server:
   - For Windows: **twh_configWP -u db2user -p <your password> -f list**
   - For AIX: **twh_configWP.sh -u db2user -p <your password> -f list**

   Example 5-7 shows the output for the IBM Tivoli NetView warehouse pack.

Example 5-7 The twh_configWP command output

```
C:\TWH\tools\bin>twh_configWP -u db2admin -p <your password> -f list
CDWCW0002I The twh_config_WP.pl program started.

Installed warehouse enablement packs:

  CODE  VERSION            ROLE  NAME
  ----  ------------------ ----  ------------------------------------------
  AMX   Version 1.1.0      V11   IBM Tivoli Monitoring
  AMY   Version 1.1.0      V11   IBM Tivoli Monitoring for Operating Systems
  ----  ------------------ ----  ------------------------------------------

Source datasources used by warehouse enablement packs:

  CODE  DATASOURCE        CLIENT_HOSTNAME
  ----  ----------------  ---------------------------------------------------
  AMX   ITM_DB            localhost
  ----  ----------------  ---------------------------------------------------

Central data warehouse datasources used by warehouse enablement packs:

  CODE  DATASOURCE        CLIENT_HOSTNAME
  ----  ----------------  ---------------------------------------------------
  AMX   TWH_CDW           localhost
  AMY   TWH_CDW           localhost
  ----  ----------------  ---------------------------------------------------

Data mart datasources used by warehouse enablement packs:

  CODE  DATASOURCE        CLIENT_HOSTNAME
  ----  ----------------  ---------------------------------------------------
  AMY   TWH_MART          localhost
  ----  ----------------  ---------------------------------------------------
CDWCW0001I The twh_config_WP.pl program completed successfully.
```

Archived

Archived

6

IBM Tivoli NetView warehouse pack

In this chapter, we discuss at what is required to provide network availability reporting in an IBM Tivoli NetView V7.1.4 and Tivoli Data Warehouse V1.3 environment using the IBM Tivoli NetView warehouse pack. We cover the following topics:

- Case study overview
- IBM Tivoli NetView warehouse pack overview
- Prerequisites
- Preparing NetView for data collection
- Installing the NetView warehouse packs
- Testing, scheduling, and promoting the ETLs
- Reporting

© Copyright IBM Corp. 2005. All rights reserved.

6.1 Case study overview

Table 6-1 gives a brief summary of the distributed IBM Tivoli Data Warehouse environment we started with to install the IBM Tivoli NetView warehouse pack.

Table 6-1 Environment for Tivoli NetView integration

Host name	Component	Database	Operating system
TDW1302	Tivoli Data Warehouse control server V1.3	DB2 Server V8.2	Microsoft Windows 2000 SP6
	Tivoli Data Warehouse central warehouse V1.3 (remote agent site)		
	Tivoli Data Warehouse data mart V1.3		
aus-nt-crystal01	Crystal Enterprise Server 10	N/A	Microsoft Windows 2000 SP6
aus-ux-netview01	IBM Tivoli NetView V7.1.4	DB2 Server V7.2 FP11 Runtime Client	AIX 5L Version 5.2 ML01

Figure 6-1 on page 125 summarizes the distributed Tivoli Data Warehouse environment used in this chapter.

Archived

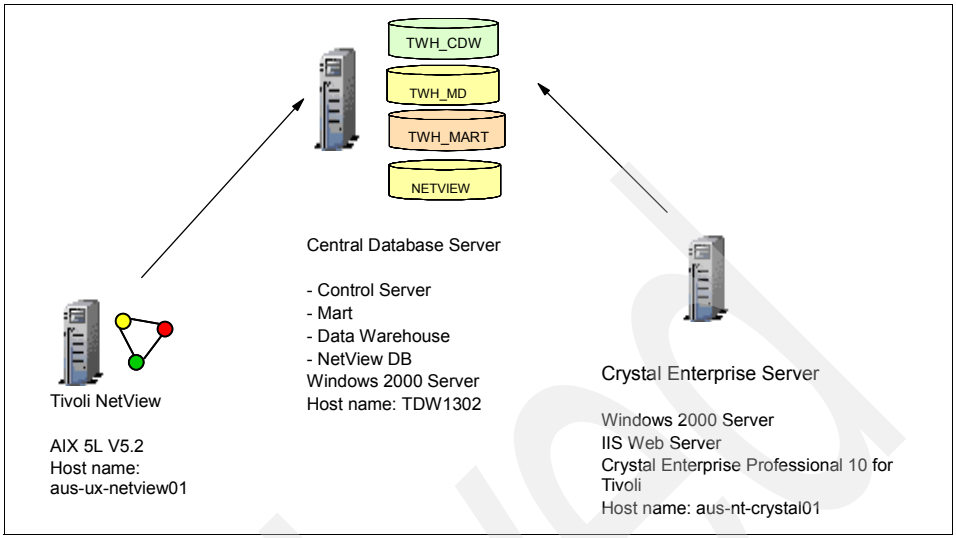


Figure 6-1 Distributed deployment scenario

6.2 IBM Tivoli NetView warehouse pack overview

The IBM Tivoli Netview warehouse pack enables Tivoli Data Warehouse reporting of network availability and performance information. The latter is provided by the IBM Tivoli Netview *snmpcol* daemon. Network node, SmartSet, Layer 2, and performance information is recorded to an availability database by IBM Tivoli Netview (hereafter referred to as NetView), which is used as input to Tivoli Data Warehouse.

> **Note:** Layer 2 information is only collected when the IBM Tivoli Switch Analyzer product is installed.

The IBM Tivoli NetView warehouse pack code is provided with the IBM Tivoli NetView V7.1.4 software.

In Figure 6-2 on page 126, the data flow of an integration of NetView in a Tivoli Data Warehouse environment is illustrated. We start with a brief description of the processes and their control.

Node availability information is stored by the NetView process *tdwdaemon* into the NetView source database. The NetView *snmpcol* daemon writes performance information gathered by SNMP polling (CPU load, number of processes, and so on) into the NetView source database. The data upload to the NetView source database is controlled by the NetView server that is illustrated in

Figure 6-2 by dashed lines. The data flow within Tivoli Data Warehouse is controlled by the Tivoli Data Warehouse control server. The generation and publishing of the NetView-specific reports is controlled by the Crystal Enterprise server.

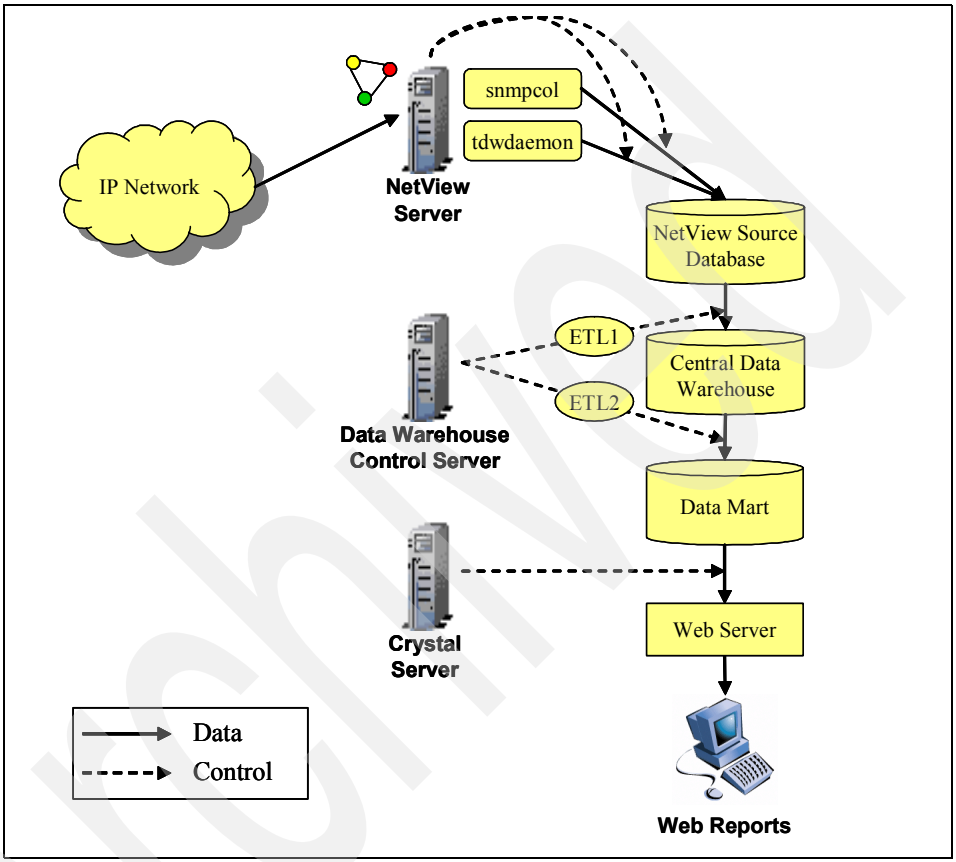


Figure 6-2 IBM Tivoli NetView warehouse pack data flow

6.3 Prerequisites

The stated prerequisites, as per the *IBM Tivoli NetView Warehouse Enablement Pack Implementation Guide*, SC32-1237, which can be found in the \tedw_apps_etl\anm\pkg\v110\doc directory of the warehouse pack software, are:

- IBM Tivoli NetView Version 7.1.4
- IBM DB2 Universal Database Enterprise Edition Version 7.2
- IBM DB2 Universal Database Enterprise Edition Version 7.2 Fix Pack 6

- IBM Tivoli Enterprise Data Warehouse required e-fixes to IBM DB2 UDE V7 Fix Pack 6 (1.1-TDW-0002)
- IBM Tivoli Enterprise Data Warehouse Version 1.1
- IBM Tivoli Enterprise Data Warehouse Version 1.1 Fix Pack 2 (1.1-TDW-FP02)

In this case study scenario chapter, we use Tivoli Data Warehouse V1.3 in a previously built distributed environment, as described in Chapter 5, “Getting Tivoli Data Warehouse V1.3 up and running” on page 91.

Here is a list of the products used and their releases for the case study scenario:

- IBM Tivoli NetView V7.1.4
- IBM Tivoli NetView V7.1.4 Fix Pack 1
- IBM DB2 Universal Database Enterprise Edition Version 7.2 Fix Pack 11 (for NetView database)
- IBM Tivoli NetView warehouse pack Version 1.1.0 Fix Pack 1
- Tivoli Data Warehouse Version 1.3 Fix Pack 2

6.3.1 Verifying prerequisites

From the stated prerequisites, we determine what action (if any) is required on our NetView server platform in our case study environment. These actions are described in Table 6-2.

Table 6-2 NetView warehouse pack prerequisite check: NetView server platform

Prerequisite	Comment	Action required
IBM Tivoli NetView V7.1.4.		Install IBM Tivoli NetView V7.1.4 Fix Pack 1.
IBM DB2 Universal Database Enterprise Edition Version 7.2 Fix Pack 11.	We use a DB2 UDB V7.2 Fix Pack 11 level on a separate server from the NetView server.	Install the DB2 Client Version 7.2 Fix Pack 11 on the NetView server machine.

6.3.2 Gathering installation information

Table 6-3 on page 128 provides the information to be used during the installation of the NetView warehouse pack.

Archived

Table 6-3 NetView warehouse pack installation information

Information required	Description	Default value	Used value
NetView availability DB name	Name of the database that will be created during installation to store the NetView availability data.	NETVIEW	NETVIEW
DB2 user ID	A DB2 user ID with *create* authority.	db2admin	dbadmin
DB2 user ID password	The DB2 user ID password.		*As required*
Data purge days	The number of days the availability data will be retained before being purged.	90	*Leave as default*
SmartSet membership retrieval time	The time of day to load SmartSet information.	23	*Leave as default*
List of SmartSets	The name of the SmartSets for which availability data will be collected. **Note:** The Router SmartSet is required.	Routers	*Add "ALL" in order to produce more data.*
Warehouse pack install directory	The directory name where the warehouse pack will be installed.	\usr\ov\dwpack	*Leave as default*
NetView availability DB name	Name of the database that will be created during installation to store the NetView availability data.	NETVIEW	NETVIEW
DB2 user ID	A DB2 user ID with *create* authority.	db2admin	db2admin
DB2 database password	The DB2 password.		

Archived

6.4 Preparing NetView for data collection

NetView uses a DB2 database as the source database for its availability and performance data. In this section, we discuss how to set up NetView to fill the source database with data.

Before performing any configuration on the NetView server, we need to perform the following steps:

- Ensure that the DB2 server to host the NetView source database is at the proper level. The minimum required by the NetView warehouse pack is IBM DB2 Universal Database Enterprise Edition Version 7.2 Fix Pack 6.
- Install and configure the DB2 client on the NetView Server machine in case the NetView source database will be placed in a separate machine. Ensure that the DB2 client version and level is compatible with the DB2 server.

6.4.1 Enabling NetView to export data for Tivoli Data Warehouse

In this section, we describe how to enable NetView to fill out its source database for use with Tivoli Data Warehouse V1.3. NetView includes an easy-to-use administration tool that:

- Creates and populates the NetView source database.
- Configures the ODBC connection to the DB2 source database.
- Modifies the SNMP Collection daemon to store performance data on the source database.
- Registers and starts the tdwdaemon, which stores network availability data to the source database.

Perform following steps to enable NetView to collect data:

1. Log in to your NetView server with administrator authority.
2. Start the installer for data export from NetView to the DB2 source database by selecting **Start** → **Programs** → **Tivoli NetView** → **Administration** → **Configure data Export to DB2 for use in Tivoli Enterprise Data Warehouse** from the Windows desktop.
3. A configuration menu, as shown in Figure 6-3 on page 130, opens. Fill in the appropriate data as collected in 6.3.2, “Gathering installation information” on page 127. Figure 6-3 on page 130 shows this menu filled with data according to our case study scenario.
4. Select **OK** to proceed. In our case study installation, we used a remote DB2 server residing on AIX host tdw009.

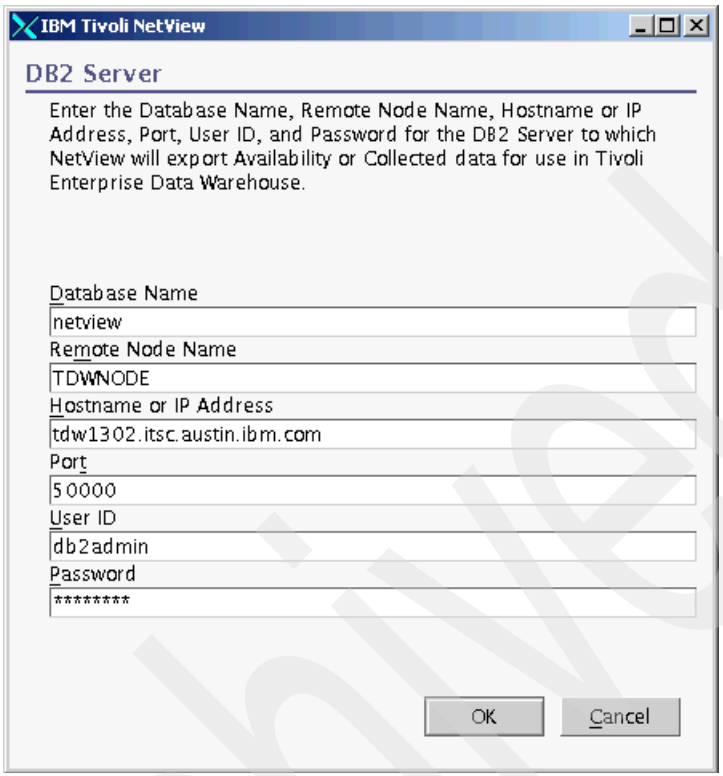


Figure 6-3 NetView configure data export to DB2: Parameters

5. Select **Create Database** in the pop-up window shown in Figure 6-4. The NETVIEW source database and its tables are created.

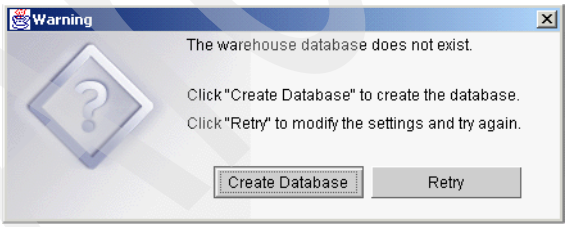


Figure 6-4 NetView configure data export to DB2: Create database

6. Select **Yes** in the pop-up window shown in Figure 6-5 on page 131 to register and start the Data Warehouse daemon (tdwdaemon).

Archived

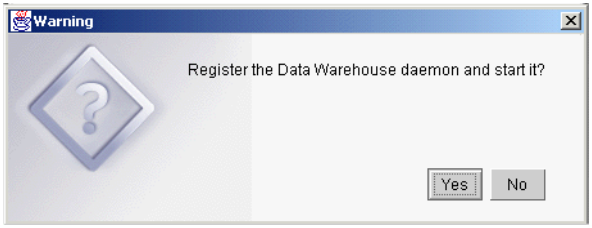


Figure 6-5 NetView configure data export to DB2: Register and start tdwdaemon

During the configuration, a number of logs might be written, depending on whether any problems are encountered during the installation. The log files we discovered are:

- /usr/OV/log/dwpack/DWP_Install_stdout.log
- /usr/OV/log/TDWError_mmddhh.log
- /usr/OV/log/tdwdaemon.log

Check these logs for unusual messages.

Verifying DB updates

We verified that data was actually being written to the availability database by issuing the following command:

```
db2 select count(*) from netview.netview_nodes
```

The greater than zero count returned (as shown in Example 6-1) indicated that data was being written.

Example 6-1 Verify NetView source database updates

```
bash-2.04$ db2 connect to NETVIEW user db2admin using db2admin

   Database Connection Information

 Database server        = DB2/NT 7.2.9
 SQL authorization ID   = DB2ADMIN
 Local database alias   = NETVIEW

bash-2.04$ db2 "select count(*) from netview.netview_nodes"

1
-----------
         94

  1 record(s) selected.
```

Archived

If the count had been zero, we could have tried the following sequence of actions:

1. Stop the tdwdaemon daemon:

   ```
   ovstop tdwdaemon
   ```

2. Stop the netmon daemon:

   ```
   ovstop netmon
   ```

3. Start the tdwdaemon daemon:

   ```
   ovstart tdwdaemon
   ```

4. Start the netmon daemon:

   ```
   ovstart netmon
   ```

5. Verify that both daemons are now active:

   ```
   ovstatus tdwdaemon
   ovstatus netmon
   ```

6. Preload the current NetView status to the availability DB:

   ```
   netmon -a 500
   ```

7. Verify that data is being written by checking the count on the netview.netview_nodes table again:

   ```
   db2 select count(*) from netview.netview_nodes
   ```

6.4.2 NetView SmartSets configuration

Some of the reports make use of NetView SmartSets. In this section, we give a short introduction to the SmartSets concept and show how to create and configure them.

SmartSets are containers that are populated with NetView objects. Dynamic or static filters define which objects are mapped to a SmartSet. You can filter by the objects' attributes.

For our Web Shop, we are creating a SmartSet based on host name patterns. Figure 6-6 on page 133 shows the SmartSet desktop of NetView. Some of the network nodes are not available. Therefore, the CriticalNodes SmartSet is displayed as red.

We now explain how to create new NetView SmartSets. However, you might need to vary the steps to meet your requirements.

Archived

From the NetView desktop, select **Tools** → **Smartset Editor** from the toolbar to open the SmartSets configuration window shown in Figure 6-6. To define a new SmartSet, select **Add** from the menu.

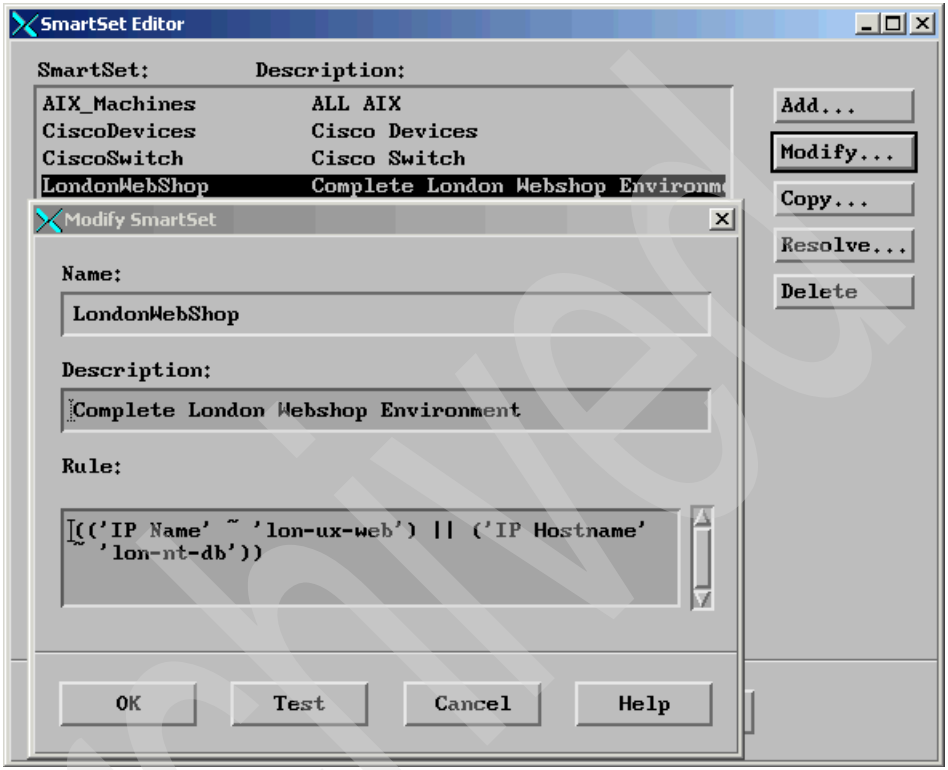


Figure 6-6 Microsoft SmartSet advanced attributes

In the **Rule** field, you can insert conditions to meet your needs. We insert the rule (refer to Figure 6-6):

```
(('IP Name' ~ 'lon-ux-web') || ('IP Hostname' ~ 'lon-nt-db'))
```

Now, the SmartSet is populated by all nodes from our Web Shop.

Archived

The new SmartSets are also displayed on the NetView SmartSets desktop, as shown in Figure 6-7.

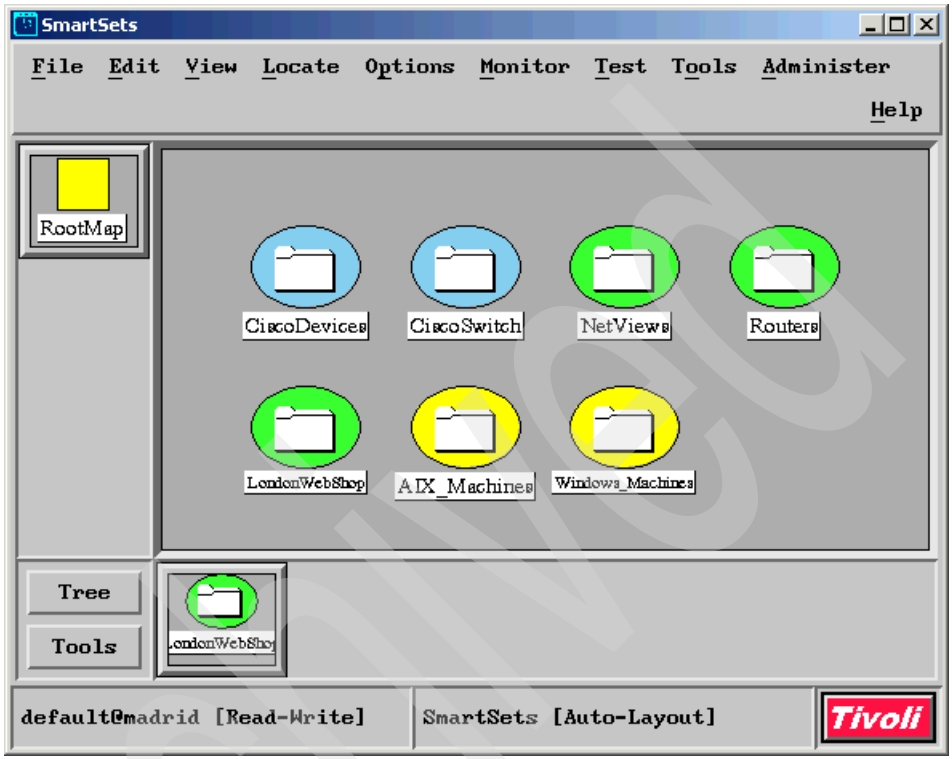


Figure 6-7 NetView SmartSets: Overview

Double-click the **LondonWebShop** SmartSet, and we will see our four machines, as shown in Figure 6-8 on page 135.

Archived

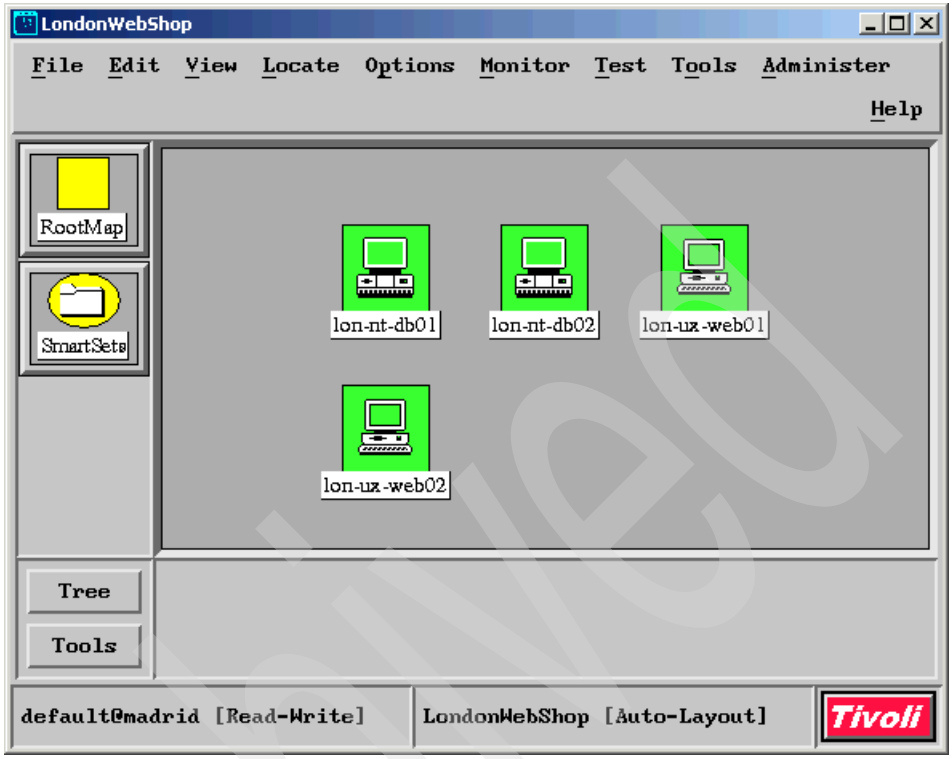


Figure 6-8 SmartSet LondonWebShop

6.4.3 Configuring NetView data warehouse daemon (tdwdaemon)

The NetView tdwdaemon fills the NetView source database with availability data from the network nodes. It also controls the expiration of data within the NetView data warehouse source database. The behavior of the tdwdaemon can be changed by editing its configuration file, which is located by default under the name /usr/OV/conf/tdwdaemon.properties.

Most of the parameters are set by the configuration process explained in 6.4.1, “Enabling NetView to export data for Tivoli Data Warehouse” on page 129 and seldom need to be changed.

Archived

The configuration parameters defined in the tdwdaemon.properties file are (see Example 6-2 on page 137):

- SMARTSETS
  Here, you can specify a comma-separated list of the NetView SmartSets you want to report. Here are some notes on this issue:
  - The SmartSet names are separated by commas, not blank spaces.
  - SmartSet names are case sensitive.
  - The default value is `Routers`.
  - The Routers SmartSet is required. If you change the SMARTSETS settings, make sure that Routers is included in your list.
  - If you specify `ALL`, the availability data of all the network nodes will be stored in the source database.
  - In our case study installation, we created new NetView SmartSets called *LondonWebShop*. We selected our self-made SmartSets along with the required SmartSet Routers, as shown in Example 6-2 on page 137, and a few other SmartSets for testing.
- SMARTSET_LOAD_TIME
  This provides the hour when the NetView SmartSet population is loaded to the NetView data warehouse source database. The default value 23 means that the data is loaded every day at 11 p.m.

  Note: We recommend that you schedule the ETL1 ANM_c05_ETL1_Process within the Tivoli data warehouse at least 1 hour after the SmartSet load time.

- OUTAGE_STORAGE_TIME
  This parameter specifies the number of days before the availability data expires in the NetView data warehouse source database. The default value is 90 days.

  Note: The SmartSet data is loaded once a day. Therefore, the NetView source database contains a snapshot of this particular point in time. SmartSets with rapid population changes, such as CriticalNodes, might not be suitable for reporting purposes.

- DBPASSWORD
  The encrypted DB2 password.

Note: To update the Db2 password, complete the following procedures:

- For UNIX:
  Run `/usr/OV/bin/serversetup` and select **Configure** → **Set options for daemons** → **Set options for topology, discovery, and database daemons** → **Set options for tdwdaemon**. The key words "Yes" and "Run Setup" need to be displayed in the "Enable tdwdaemon" and "DB2 Connection" drop-down lists. Then, click **OK** or **Apply**. Click **Yes** to continue and enter the new DB2 password. After seeing the message `"The warehouse database already exists."`, click **Save** to complete the update.
- For Windows:
  Click **Start** → **Programs** → **Tivoli NetView** → **Administration** → **Configure Data Export to DB2 for use in Tivoli Enterprise Data Warehouse**. Enter the new password and click **OK**. After seeing the message "`The warehouse database already exists.`", click **Save** to update the password.

- DB2USER
  The DB2 user ID (DB2 administrator or any user ID with create authority).
- DBNAME
  The name of the NetView source database. The default is NETVIEW.
- Port
  DB2 database IP-port of the NetView source database. The default is 50000.
- HOSTNAME
  Host name of the server that hosts the NetView source database.
- NODENAME
  DB2 database node of the NetView source database. The default is TDWNODE.

Example 6-2 NetView tdwdaemon configuration file tdwdaemon.properties

```
#Mon Aug 23 10:58:26 CDT 2004
PORT=50000
HOSTNAME=tdw1302
DBUSER=db2admin
SMARTSET_LOAD_TIME=14
NODENAME=TDWNODE
SMARTSETS=ALL,Routers,LondonWebShop,AIX_Machines,Windows_Machines
DBPASSWORD=ZGIyYWRtaW4\=
OUTAGE_STORAGE_TIME=90
DBNAME=netview
```

After changing the tdwdaemon.properties file, you have to restart the tdwdaemon daemon to apply the changes. Use the NetView commands **ovstop** and **ovstart** for this purpose. Example 6-3 shows the command shell dialog.

Example 6-3 Restart the NetView data warehouse daemon tdwdaemon

```
[root@aus-ux-netview01:/usr/OV/bin] ovstop tdwdaemon
[root@aus-ux-netview01:/usr/OV/bin] ovstart tdwdaemon
```

Verifying NetView data collection enablement

In order to verify that the data collection is taking place, perform these tasks:

- Check the tdwdaemon daemon.
- Check the snmpcollect daemon.
- Check the existence of data in the NetView source database.

Verifying NetView data warehouse daemon (tdwdaemon)

The NetView daemon named tdwdaemon is used to perform availability data recording. We verify that this daemon started successfully by using the following command:

```
ovstatus tdwdaemon
```

Example 6-4 shows the command shell output executing this command.

Example 6-4 Status of NetView data warehouse daemon (tdwdaemon)

```
[root@aus-ux-netview01:/usr/OV/bin] ovstatus tdwdaemon
 object manager name: tdwdaemon
 behavior:            OVs_WELL_BEHAVED
 state:               RUNNING
 PID:                 24050
 last message:        Initialization complete
 exit status:         -

[root@aus-ux-netview01:/usr/OV/bin]
```

The log file for the tdwdaemon daemon is named *tdwdaemon.log* and can be found in the /usr/OV/log directory. Verify that no apparent errors were being reported on the tdwdaemon.log file. You will also find DB2 errors in this log file that tdwdaemon has encountered while communicating with the NetView data warehouse source database.

Note: If no tdwdaemon.log exists for tdwdaemon to write to, it will create a new one. Deleting it prior to restarting the tdwdaemon daemon makes it easier to review, because all the old entries are removed.

Verifying NetView SNMP collector daemon (snmpcollect)

Performance data is stored to the NetView data warehouse source database through the NetView SNMP collector daemon named *snmpcollect*. We verify that this daemon started successfully by using the following command:

```
ovstatus snmpcollect
```

Example 6-5 shows the command shell output executing this command.

Example 6-5 Status of the NetView SNMP collector daemon (snmpcollect)

```
[root@aus-ux-netview01:/usr/OV/bin] ovstatus snmpcollect
 object manager name: tdwdaemon
 behavior:            OVs_WELL_BEHAVED
 state:               RUNNING
 PID:                 24050
 last message:        Initialization complete
 exit status:         -

[root@aus-ux-netview01:/usr/OV/bin]
```

The log file for the tdwdaemon is named *snmpCol.trace* and can be found in the /usr/OV/log directory. Verify that no apparent errors were being reported on the snmpCol.trace log file in the directory /usr/OV/log. You will also find DB2 errors in this log file that the snmpcollect daemon has encountered while communicating with the NetView data warehouse source database.

Verifying data import to the NetView source database

To verify whether data is imported to the NetView source database, you can use the DB2 Control Center to view the contents of the source databases tables. However, we give an example using the DB2 command line. Log in to the NetView database using the command:

```
db2 connect to NETVIEW user db2inst1 using <password>
```

Where `<password>` is your password for the db2inst1 database user.

Here is a list of the items to check and the commands to use to check them:

- Availability data:

  ```
  db2 select count(*) from netview.netview_nodes
  ```

- Performance data:

  ```
  db2 select count (*) from netview.snmpcollection
  ```

- NetView SmartSets:

  ```
  db2 select count (*) from netview.smartsets
  ```

Archived

In all three cases, the count must be greater than 0. Example 6-6 shows the results of these commands for our test study environment.

> **Note:** You might have to wait until the SmartSet data upload has taken place. The time for this upload is specified in the configuration file tdwdaemon.properties.

Example 6-6 Check the NetView source database

```
$ db2 connect to netview user db2admin using db2admin

   Database Connection Information

 Database server        = DB2/NT 8.1.5
 SQL authorization ID   = DB2ADMIN
 Local database alias   = NETVIEW

$ db2 "select count(*) from netview.smartsets"

1
-----------
          5

  1 record(s) selected.
$
```

6.5 Installing the NetView warehouse packs

As described in 6.2, “IBM Tivoli NetView warehouse pack overview” on page 125, NetView can collect availability and performance information into the NetView data source database. In order to gather this data into the Tivoli Data Warehouse V1.3 environment, the following two warehouse packs must be installed:

- NetView availability warehouse pack (ANM)
- NetView SNMP performance warehouse pack (AN1)

In this section, we explain how to install and configure both of these warehouse packs.

6.5.1 Backing up the Tivoli Data Warehouse environment

We strongly advise that you back up your data warehouse environment prior to any installation activity.

Archived

In order to back up the Tivoli Data Warehouse V1.3 environment, stop the DB2 Warehouse logger service (vwlogger) and back up the control server database (TWH_MD), all central data warehouse databases, and all data mart databases.

For example, in order to back up the TWH_MD, TWH_CDW, and TWH_MART databases, perform the following commands:

```
db2 backup database TWH_MD to C:\\backup
db2 backup database TWH_CDW to C:\\backup
db2 backup database TWH_MART to C:\\backup
```

6.5.2 Establishing ODBC connections

In this section, we show how to set up database connections to the NetView source database and how to configure the data warehouse sources and targets to make use of the ODBC data sources.

Note: We used NETVIEW as the ODBC data source name. This is different from what is used in *IBM Tivoli NetView Warehouse Enablement Pack Implementation Guide*, SC32-1237, which uses ANM_SOURCE as the ODBC data source name. AIX DB2 instances can only use a maximum of eight characters in names for database aliases. ANM_SOURCE has 10 characters and therefore cannot be used on AIX DB2 instances.

Creating a NETVIEW ODBC data source

By default, the data warehouse agent on the data warehouse control server is used to execute the NetView ETLs. Therefore, you have to create the ODBC data source on the data warehouse control server. However, if you plan to execute the ETLs or parts of them on a remote data warehouse agent, you must also create an ODBC data source for the NetView source database on the remote agent site.

In order to create a NetView ODBC data source for the NetView source database, perform the following steps:

1. Log in to the data warehouse control server or the remote agent site server with administrator authority and select **Start** → **Control Panel** → **Administrative Tools** → **Data Sources (ODBC)**.
2. Then, select **System DSN** to view ODBC data source administration window, as shown on the left side in Figure 6-9 on page 142.
3. Select **Add** to open the Create New Data Source window, as shown on the right side in Figure 6-9 on page 142. Here, you select the **IBM DB2 ODBC DRIVER**. Then, select **Finish**.

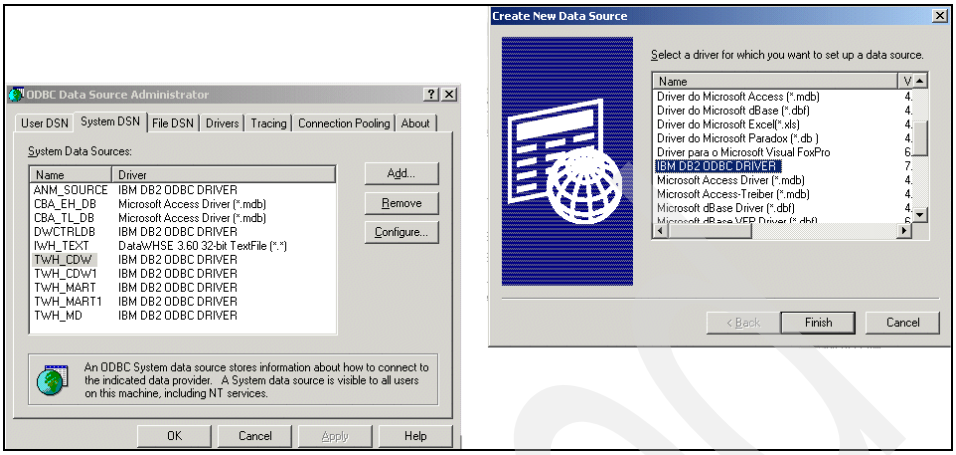


Figure 6-9 Create an ODBC data source for NETVIEW

4. Figure 6-10 shows the ODBC IBM DB2 Driver - Add window. You can add an optional description. If your NETVIEW source database is local or already cataloged within the DB2 client, you need to select **NETVIEW** from the Database alias dialog box.

   If you use a remote database, select **Add**. This opens the Add Database Wizard window.

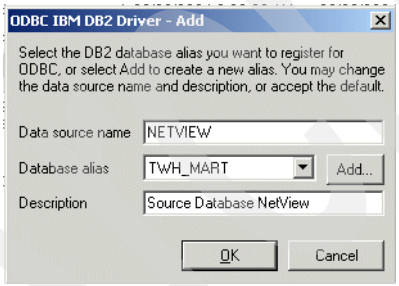


Figure 6-10 Add an ODBC data source for NETVIEW

   Here are the different steps we have to perform in the Add Database Wizard (these steps are also shown in Figure 6-11 on page 143):

   a. Source
      We selected **Manually configure a connection to a database**.

   b. Protocol
      We selected **TCP/IP**.

   c. TCP/IP
      We inserted data, as shown in Table 6-4 on page 143.

Table 6-4 Add database wizard: Register TCP/IP

Selection	inserted values	remark
Host name	`tdw1302.itsc.austin.ibm.com`	Host name of the database server
Port number	`50000`	Default DB2 service port
Service name		Optional, left blank

d. Database
We inserted `NETVIEW` as the database name. Select **Finish** to start the registration.

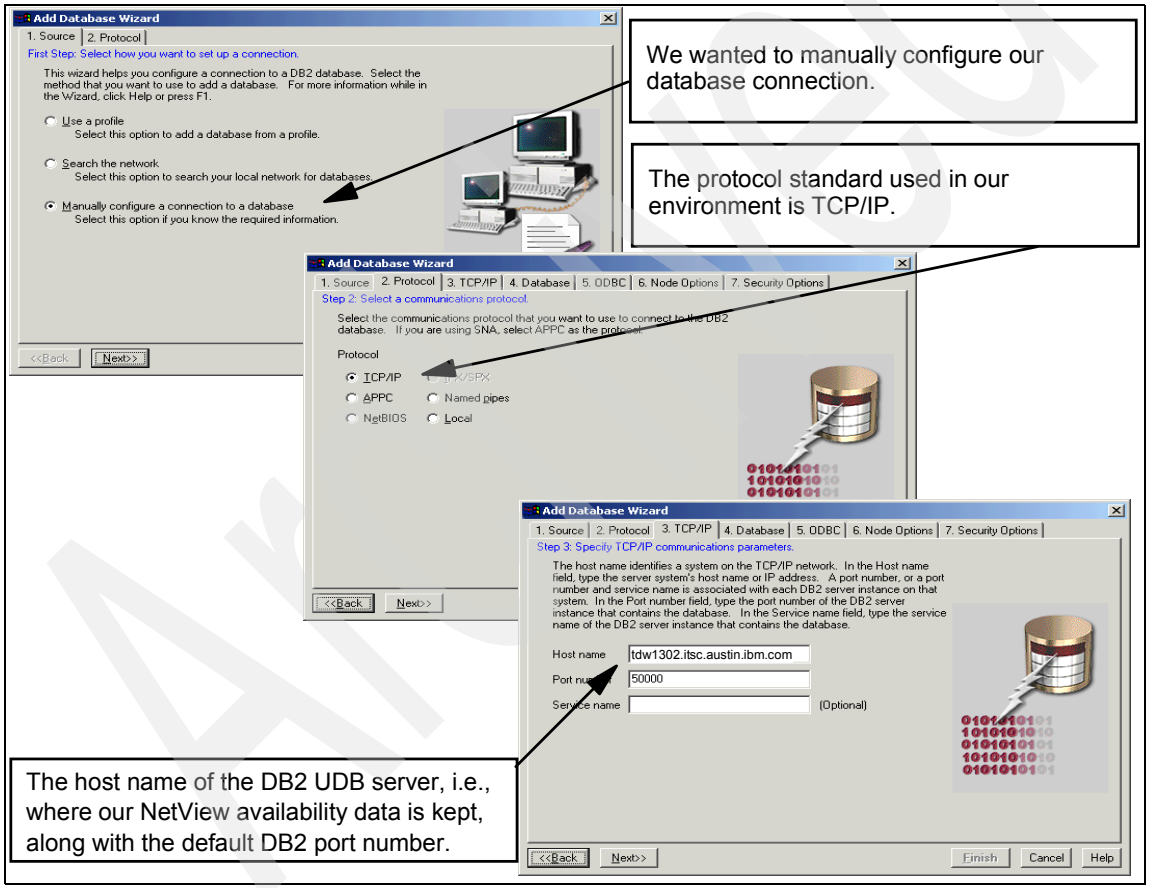


Figure 6-11 Configure NetView source database connectivity

6.5.3 Installing NetView warehouse pack software

In this section, we describe how we installed the NetView availability warehouse pack (ANM) in our case study scenario environment. The installation of the NetView SNMP performance warehouse pack (AN1) can be performed in a similar way.

Warehouse pack installations must be performed on the control server machine.

In order to install the NetView availability warehouse pack (ANM), perform the following steps:

1. While logged on to the control server as a user ID with administrator authority, select **Start** → **Programs** → **Tivoli Data Warehouse** → **Install a Warehouse Pack.** The InstallShield wizard is started.
2. On the Welcome window, select **Next** to continue. In the next window, the path to your Common Logging Directory is displayed. In the case study installation, the path is C:\Program Files\ibm\tivoli\common.
3. The InstallShield wizard checks the Tivoli Data Warehouse environment. This might take a few minutes. In the next window, you can select the warehouse packs to install. This list is empty, as shown in Figure 6-12.

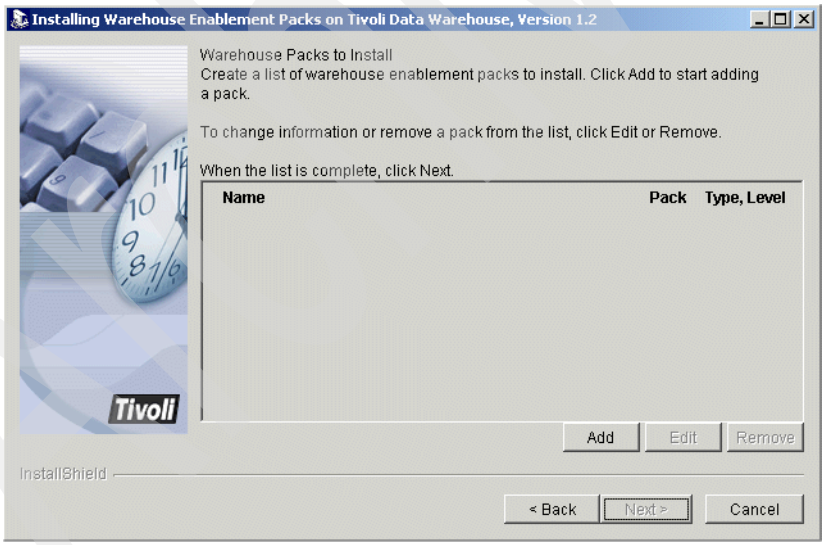


Figure 6-12 NetView warehouse pack installation: List of WPs to install

4. Select **Add** and insert the location of the NetView warehouse pack installations properties file. This file is named twh_install_props.cfg. The properties file for NetView availability warehouse pack with code ANM can be found on the installation media under the \tedw_apps_etl\anm\pkg\ directory, as shown in Figure 6-13. The properties file for NetView SNMP performance warehouse pack with code AN1 can be found on the installation media under the \snmp_etl\an1\pkg\ directory.

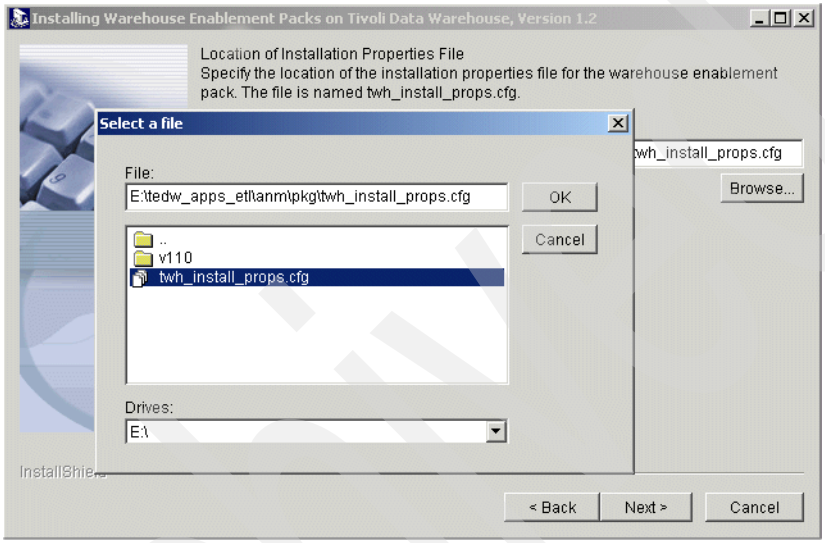


Figure 6-13 NetView warehouse pack installation: Properties file

5. Select **OK** or **Next**, or both, to get back to List of Warehouse Packs to install. In contrast to Figure 6-12 on page 144, the list is now populated with the NetView warehouse pack, as shown in Figure 6-14 on page 146.

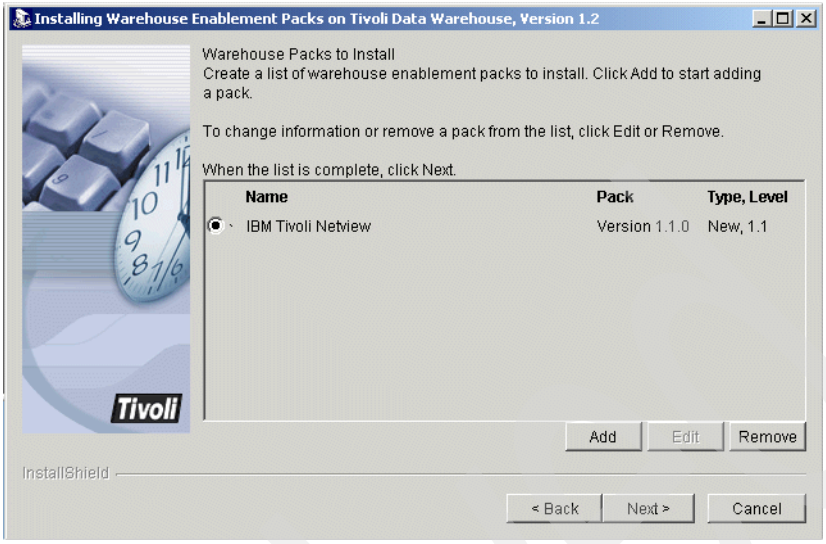


Figure 6-14 NetView warehouse pack installation: List of WPs to install NetView

6. Select **Next** to proceed with the installation. The installation takes a few minutes. Figure 6-15 shows the window that opens after a successful installation.

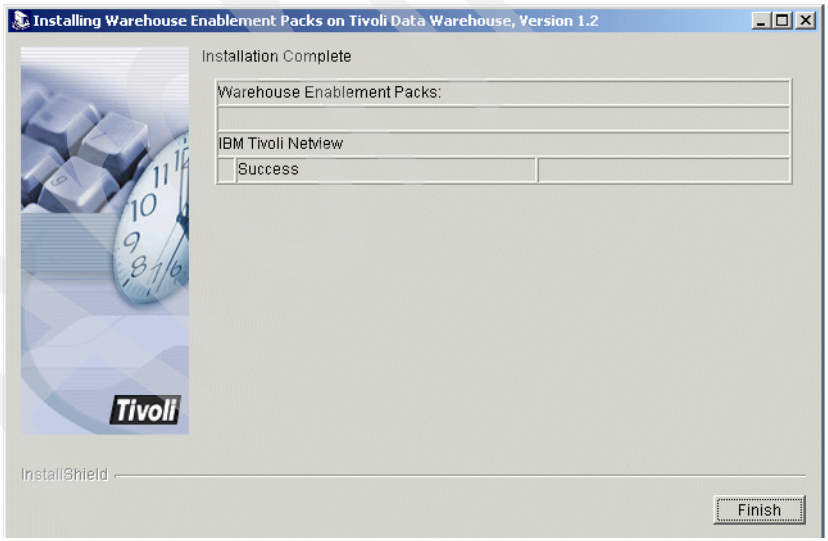


Figure 6-15 NetView warehouse pack installation: Successful installation

To install both NetView warehouse packs, you have to perform the installation steps twice using the different properties files of the two warehouse packs.

6.5.4 Fix Pack 1 for IBM Tivoli NetView warehouse pack

Fix Pack 1 for IBM Tivoli NetView warehouse pack (ANM) V1.1.0 for availablity data is bundled with NetView V7.1.4 Fix Pack 1 and fixes several issues with the ANM warehouse pack.

The installation for this fix pack is no classic warehouse pack installation, but rather a step-by-step procedure that has to be performed manually:

1. Ensure that no ETLs are currently running.
2. Execute the **`1.1.0-anm-fp01.exe`** command by double clicking it.
3. When install is complete, in the Data Warehouse Center, select **Subject Areas** → **ANM_IBM_Tivoli_Netview_v1.1.0_Subject_Area** → **Processes** → **ANM_c05_ETL1_Process**.
4. On the right side, select **ANM_c05_s010_extractNodeInfo** through **ANM_c05_s030_loadNodeInfo**. Right-click and select **Mode** → **Development**. Allow the mode to change.
5. On the right side, select **ANM_c05_s010_extractNodeInfo** through **ANM_c05_s030_loadNodeInfo**. Right-click and select **Mode** → **Production**. Allow the mode to change.
6. Use the **`db2cmd`** command to open a DB2 command prompt. Execute the command:

   ```
   db2 connect to TWH_CDW user <db2user> using <db2passwd>
   ```

   Where `<db2user>` is the DB2 user ID that initially installed the NetView warehouse pack, and `<db2passwd>` is the password for that user.
7. Navigate to the directory %TWH_TOPDIR%\apps\anm\v110\misc\.
8. Execute the command:

   ```
   db2 -td$ -f RUN_FOR_CDW.SQL
   ```
9. After completion, execute the command:

   ```
   db2 connect to TWH_MART user <db2user> using <db2passwd>
   ```

   Where `<db2user>` is the DB2 user ID that initially installed the NetView warehouse pack, and `<db2passwd>` is the password for that user.
10. Execute the command:

    ```
    db2 -td$ -f RUN_FOR_MART.SQL
    ```
11. In the Data Warehouse Center, select **Subject Areas** → **ANM_IBM_Tivoli_Netview_v1.1.0_Subject_Area** → **Processes** → **ANM_m05_ETL2_Process**.

Archived

12. On the right side, select **ANM_m05_s010_metric** through **ANM_m05_s070_total**. Right-click and select **Mode** → **Development**. Allow the mode to change.
13. On the right side, select **ANM_m05_s010_metric** through **ANM_m05_s070_total**. Right-click and select **Mode** → **Production**. Allow the mode to change.

In addition, there are some additional notices that have to be consulted: defining the authority to the warehouse sources and targets.

Additional notes

You need to also consider the following additional items (see Example 6-7):

- To ensure complete functionality, the fix for IY55671, to be installed on the NetView server, is also required. This fix comes with NetView V7.1.4 Fix Pack 1.
- When running the RUN_FOR_CDW.SQL and RUN_FOR_MART.SQL commands, the following errors might be received and can be disregarded: “`SQL0204N  "<TABLENAME>"` is an undefined name.
- When using the 1.1.0 Version of the NetView warehouse pack with the 1.3 Version of Tivoli Data Warehouse, the following error might be received during the initial run of the ETLs, during the step in line 3 of the ETL 2 process (see Example 6-7):

  ```
  RPI.SS_JOINCOLS cannot be used because it has been marked inoperative.
  SQLSTATE=51024
  ```

  If you receive this error, restart the ETL 2 process from the step in line 3. It will recreate view PRI.SS_JONCOLS.

Example 6-7 SQL commands

```
connect to twh_md user <userid> using <password>
drop view rpi.ss_joincols
create view RPI.ss_joincols as
SELECT
ss.name as ssname,
a1.largedrschemaname as schema,
a1.largedrtablename as tabname,
a1.largename as colname
FROM
IWH.AttributeLink at,
IWH.Relationship r1,
IWH.StarSchema ss,
IWH.Relationship r2,
IWH.attribute a1
where
```

```
at.type='JOIN' and
at.iwhid = r1.source_iwhid and
r1.relation_name = 'StarSchema_TO_AttLnk' and
r1.target_iwhid = ss.iwhid and
r2.relation_name = 'AttLink1_TO_Attr_Rel' and
r2.source_iwhid = at.iwhid and
a1.iwhid = r2.target_iwhid
union
SELECT
ss.name as ssname,
a1.largedrschemaname as schema,
a1.largedrtablename as tabname,
a1.largename as colname
FROM
IWH.AttributeLink at,
IWH.Relationship r1,
IWH.StarSchema ss,
IWH.Relationship r2,
IWH.attribute a1
where
at.type='JOIN' and
at.iwhid = r1.source_iwhid and
r1.relation_name = 'StarSchema_TO_AttLnk' and
r1.target_iwhid = ss.iwhid and
r2.relation_name = 'AttLink2_TO_Attr_Rel' and
r2.source_iwhid = at.iwhid and
a1.iwhid = r2.target_iwhid;
```

The IBM Tivoli NetView warehouse pack adds NetView-specific data warehouse sources and targets to the Tivoli Data Warehouse environment. After the initial installation of the NetView warehouse packs, these sources and targets need to be configured. (Refer to *IBM Tivoli NetView Warehouse Enablement Pack Implementation Guide*, SC32-1237.) The implementation guide suggests using ANM_SOURCE as the name for the data source. In our case study, we preferred NETVIEW for this purpose, as we explain in 6.5.2, “Establishing ODBC connections” on page 141.

The IBM Tivoli NetView warehouse pack includes two warehouse packs: the availability warehouse pack with code ANM and the SNMP performance warehouse pack with code AN1. The latter does not include ETL2, data mart structure, or reports. Therefore, neither the data mart target nor the central data warehouse source exists for this warehouse pack.

Table 6-5 on page 150 shows an overview of all the NetView-based sources and targets. The last column shows the appropriate user IDs for our case study installation. The databases are running on a Windows 2000 platform with the DB2 default user db2admin.

Table 6-5 NetView sources and targets

Name in data warehouse	Description	Database	Used by	User
ANM_AVAIL_Source	ODBC data source for the NetView source database	NETVIEW	ETL1	db2admin
ANM_TWH_CDW_Source	ODBC data source for the central data warehouse	TWH_CDW	ETL2	db2admin
ANM_TWH_CDW_Target	ODBC data target for the central data warehouse	TWH_CDW	ETL1	db2admin
ANM_TWH_MART_Target	ODBC data target for the data mart	TWH_MART	ETL2	db2admin
ANM_TWH_MD_Target	ODBC data target control server database	TWH_MD	ETL2	db2admin
AN1_SNMP_Source	ODBC data source for the NetView source database	NETVIEW	ETL1	db2admin
AN1_TWH_CDW_Target	ODBC data source for the NetView warehouse source database	TWH_CDW	ETL1	db2admin

To configure the Tivoli Data Warehouse Sources and targets for NetView, perform the following steps:

1. Open the DB2 Control Center by selecting **Start** → **Programs** → **IBM DB2** → **Control Center**.
2. From the DB2 Control Center, open the DB2 Data Warehouse Center by selecting **Tools** → **Data Warehouse Center** from the toolbar.
3. In the Data Warehouse Center Logon window, type the user ID of the data warehouse administrator (default to db2admin) and the appropriate password. Select **Advanced** to ensure that the control database is set to `TWH_MD`, as shown in Figure 6-16 on page 151.

Archived

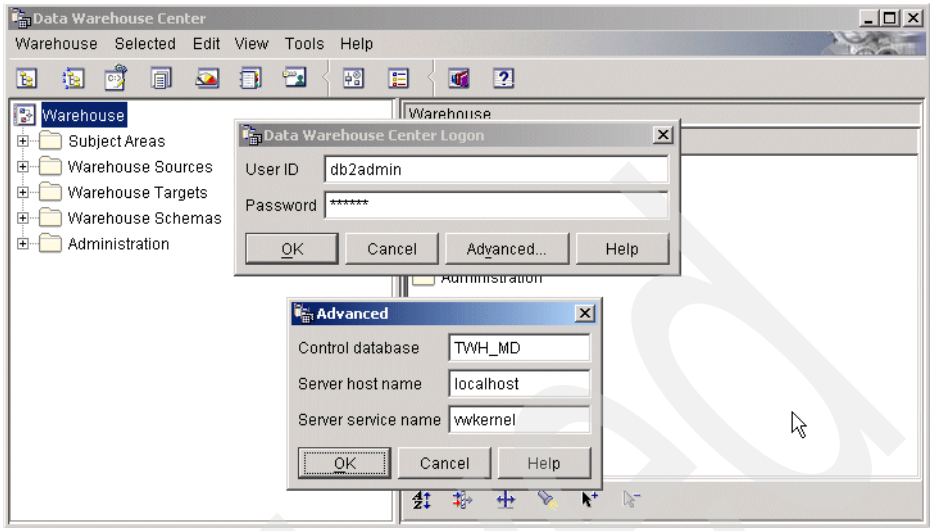


Figure 6-16 Data Warehouse Control Center: Check control database

If you have opened the Data Warehouse Control Center, you will see a browser tree, as shown in Figure 6-17 on page 152. There are, among others, the leaves *Warehouse Sources* and *Warehouse Targets*. In this section, we first discuss the configuration of the data warehouse authorities for the data warehouse sources and then the data warehouse targets for use with the NetView warehouse packs.

Authority for data warehouse sources

To configure the data warehouse sources, complete the following steps:

1. Open the **Warehouse sources** tree and select **Properties** in the context menu of a source. As shown in Figure 6-17 on page 152, we selected **ANM_AVAIL_Source** for our example.
2. In the Properties window of the selected data warehouse source, select the **Data Source** tab, also shown for ANM_AVAIL_Source in Figure 6-17 on page 152.
3. For the Data source name, select the specific data source for your environment. For our case study installation, the data sources for the NetView data warehouse sources are listed in Table 6-5 on page 150. Therefore, we entered `NETVIEW` as the data source for our ANM_AVAIL_Source example in Figure 6-17 on page 152.
4. Insert the user ID and appropriate password for the NetView source database. In our case study, we used a DB2 database on Windows for which the default user ID is `db2admin`, as shown in Figure 6-17 on page 152.
5. Select **OK** to finish the data warehouse source configuration.

Repeat these steps for all NetView data warehouse sources listed in Table 6-5 on page 150.

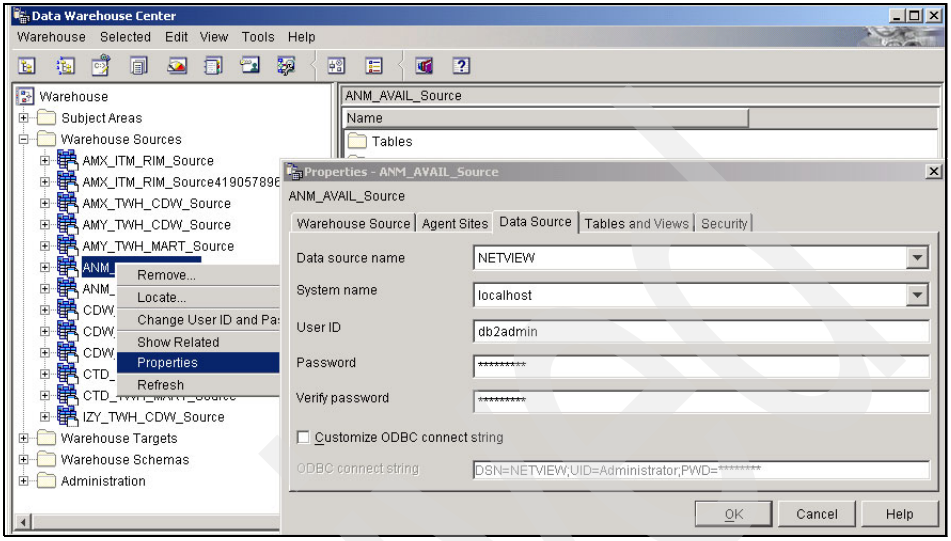


Figure 6-17 Configure NetView data warehouse sources

Authority for data warehouse targets

For the NetView data warehouse target configuration, open the Data Warehouse Control Center. Starting with the Control Center, perform the following steps:

1. Open the **Warehouse targets** tree and select **Properties** in the context menu of a target. In Figure 6-18 on page 153, for example, we choose **ANM_TWH_CDW_Target**.
2. In the Properties window of the selected data warehouse target, select the **Database** tab, as shown for ANM_TWH_CDW_Target in Figure 6-18 on page 153.
3. For the Database name, select the specific database for your environment. For our case study installation, the databases for the NetView data warehouse targets are listed in Table 6-5 on page 150. `TWH_CDW` was already configured as needed, so we can leave it as is for our case study (refer to Figure 6-18 on page 153).
4. Insert the user ID and appropriate password to the target database. Because in our case study the TWH_CDW database is on a Windows machine, we used `db2admin`, as shown in Figure 6-18 on page 153.
5. Select **OK** to finish the data warehouse source configuration.

Repeat these steps for all NetView data warehouse targets listed in Table 6-5 on page 150.

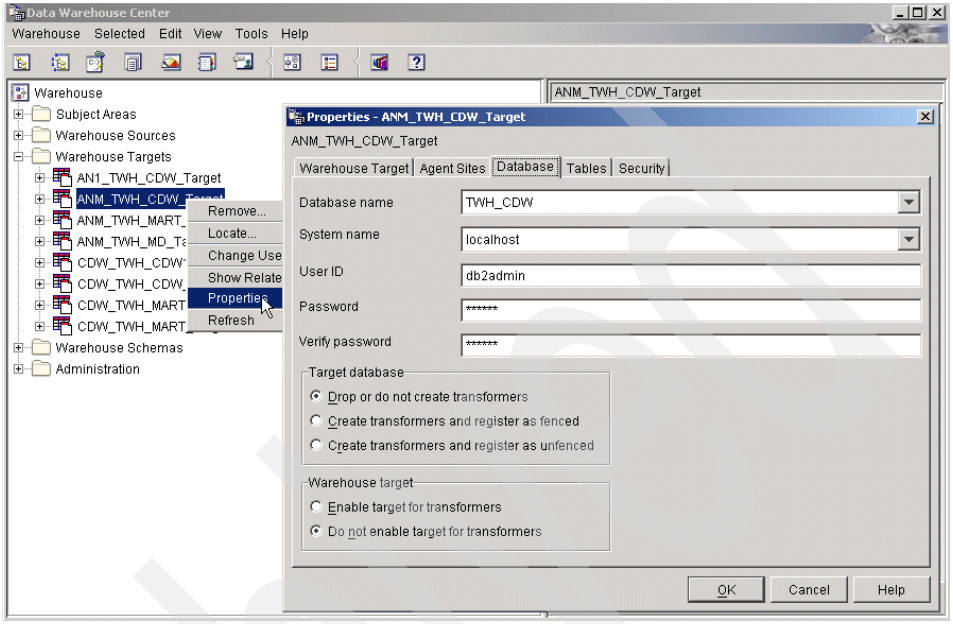


Figure 6-18 Configure NetView data warehouse targets

6.6 Testing, scheduling, and promoting the ETLs

After successfully installing and configuring the NetView warehouse packs, all the ETL1 and ETL2 processes can now be tested, scheduled, and promoted to Production mode.

There are three known modes for ETL processes in Tivoli Data Warehouse:

- Development: In this mode, process steps can be changed and their schedule can be configured. However, they do not run on their scheduled times and they cannot be tested.
- Test: In this mode, process steps are not scheduled, but they can be tested and their schedule can be changed.
- Production: In this mode, the processes run as scheduled. Neither the process steps nor their schedules can be changed.

6.6.1 Promoting the ETLs to Test mode

In order to test ETLs, we must promote them to Test or Production mode first. For our case study environment, we choose the Test mode.

On the control server machine, open the Data Warehouse Control Center and select **Subject Areas** → **ANM_IBM_Tivoli_Netview_v1.1.0_Subject_Area** → **Processes** → **ANM_c05_ETL1_Process** to open the list with the process steps. Right-click all the selected **ANM_c05_ETL1_Process** processes and select **Mode** → **Test** in its context menu, as shown in Figure 6-19.

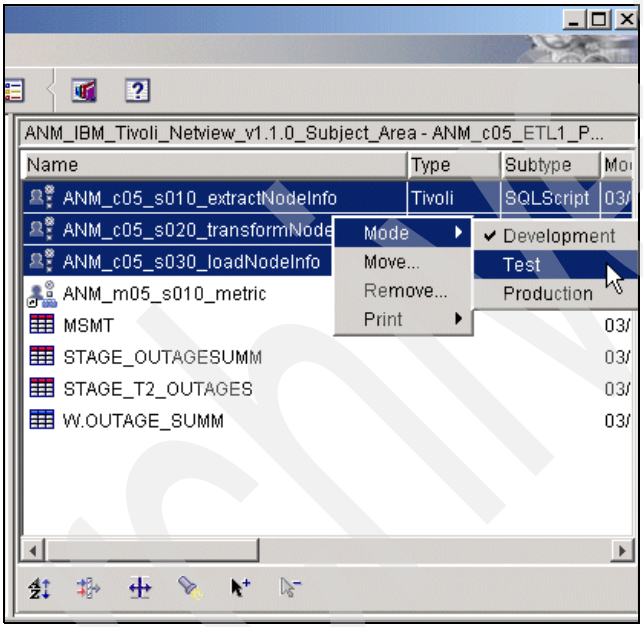


Figure 6-19 Promote ETLs to Test mode

You have to promote all process steps you want to test into Test mode.

6.6.2 Testing the ETLs

You can test the ETLs manually first before scheduling and promoting to Production mode. After you have promoted the process steps to Test mode, right-click the **ANM_c05_s010_extractNodeInfo** process and select **Test**, as shown in Figure 6-20 on page 155. The process step is executed immediately.

Archived

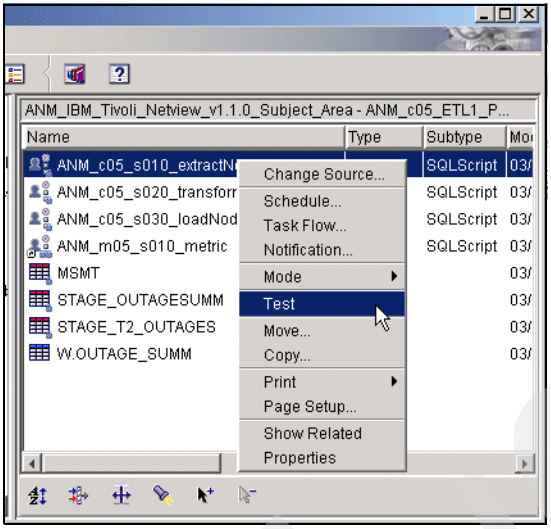


Figure 6-20 Test ETL process steps

To view the results, select **Warehouse** → **Work in progress** from the Data Warehouse Control Center. The Work in Progress window opens displaying a line for each executed process step, as shown in Figure 6-21. You can right-click the process and select **Show Log** from the context menu to open the log window. There, you can see additional information regarding the process step execution. In case of failure, this is where you will find the error messages.

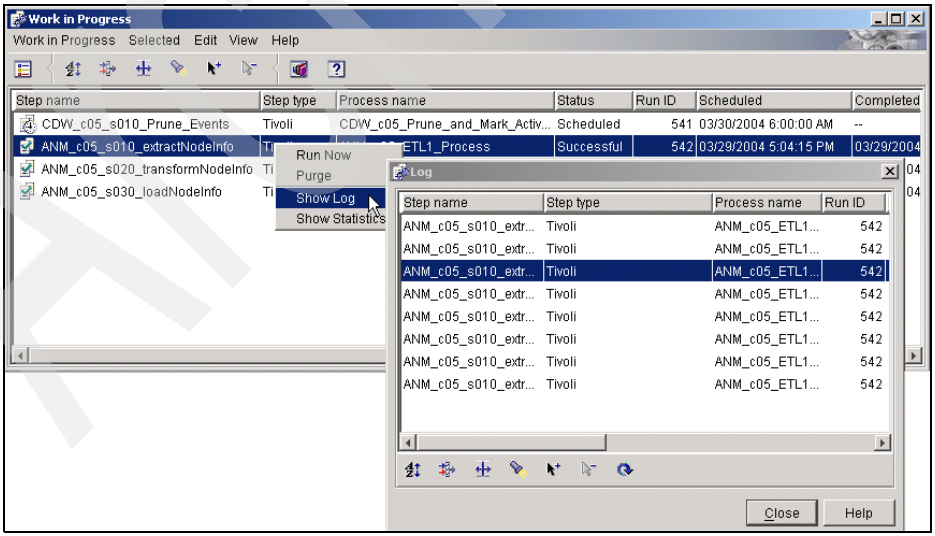


Figure 6-21 Work in Progress: Log file

Important: Do not change the sequence of the processing steps. You might cause great damage to the data within the Tivoli Data Warehouse databases.

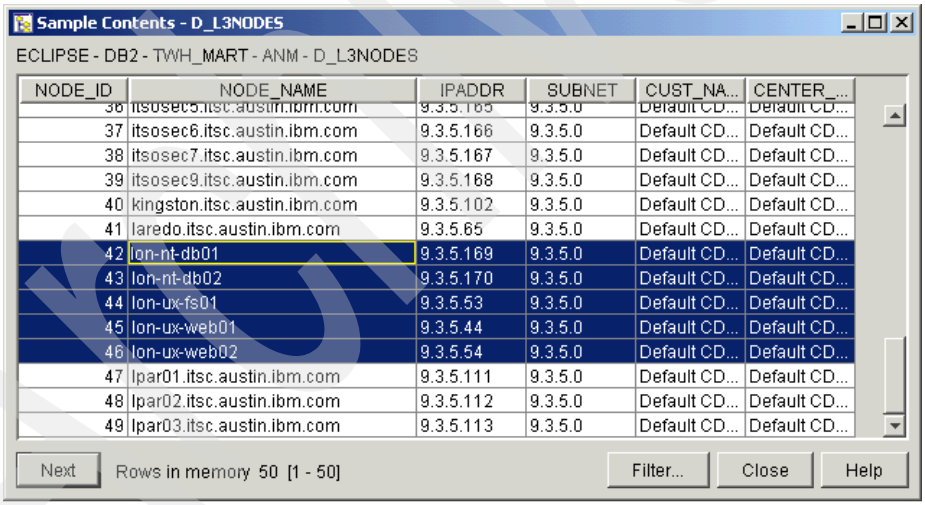
ETLs: Data collection verification

After all the processes are tested, you need to verify data collection on the Tivoli Data Warehouse databases: central data warehouse and data mart.

There are several ways to accomplish this task. Here, we show one of them:

1. On the control server machine, open the DB2 Control Center by selecting **Start** → **Programs** → **DB2** → **Control Center**.
2. Double-click the desired database, for example, the TWH_MART database.
3. Double-click **Tables**; on the right-side pane, all the tables are displayed. In the case of the TWH_MART database, select the **D_L3NODES** table. Right-click and select **Sample Contents**.

   You will see a window, as shown in Figure 6-22, displaying the sample content. This verifies that your ETL executions are successful.



NODE_ID	NODE_NAME	IPADDR	SUBNET	CUST_NA...	CENTER_...
36	itsosec5.itsc.austin.ibm.com	9.3.5.165	9.3.5.0	Default CD...	Default CD...
37	itsosec6.itsc.austin.ibm.com	9.3.5.166	9.3.5.0	Default CD...	Default CD...
38	itsosec7.itsc.austin.ibm.com	9.3.5.167	9.3.5.0	Default CD...	Default CD...
39	itsosec9.itsc.austin.ibm.com	9.3.5.168	9.3.5.0	Default CD...	Default CD...
40	kingston.itsc.austin.ibm.com	9.3.5.102	9.3.5.0	Default CD...	Default CD...
41	laredo.itsc.austin.ibm.com	9.3.5.65	9.3.5.0	Default CD...	Default CD...
42	lon-nt-db01	9.3.5.169	9.3.5.0	Default CD...	Default CD...
43	lon-nt-db02	9.3.5.170	9.3.5.0	Default CD...	Default CD...
44	lon-ux-fs01	9.3.5.53	9.3.5.0	Default CD...	Default CD...
45	lon-ux-web01	9.3.5.44	9.3.5.0	Default CD...	Default CD...
46	lon-ux-web02	9.3.5.54	9.3.5.0	Default CD...	Default CD...
47	lpar01.itsc.austin.ibm.com	9.3.5.111	9.3.5.0	Default CD...	Default CD...
48	lpar02.itsc.austin.ibm.com	9.3.5.112	9.3.5.0	Default CD...	Default CD...
49	lpar03.itsc.austin.ibm.com	9.3.5.113	9.3.5.0	Default CD...	Default CD...

Figure 6-22 Sample contents

6.6.3 Scheduling the ETLs

There are two processes that need to be scheduled for the NetView warehouse pack:

- ANM_c05_ETL1_Process
- AN1_c05_SNMP_ETL1_Process

Note: Do not schedule the ANM_m05_ETL2_Process to run. This process is automatically started by the ANM_c05_ETL1_Process.

The following steps are similar for both processes, and we use ANM_c05_ETL1_Process as an example:

1. On the Tivoli Enterprise Data Warehouse Control Center server using expand **Subject Areas**, select **ANM_IBM_Tivoli_Netview_v1.1.0_Subject_Area** → **Processes** and right-click **AMN_c05_ETL1_Process**.
2. Choose **Schedule**, as shown in Figure 6-23.

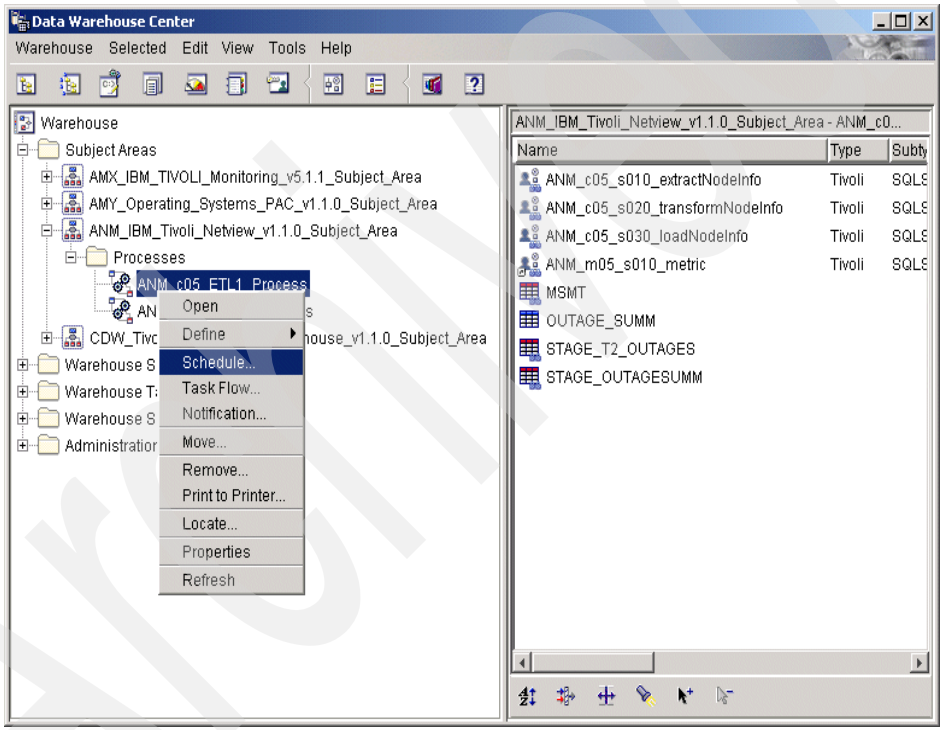


Figure 6-23 Schedule ANM_c05_ETL1_Process

3. Selecting **Schedule** will open up a window, as shown Figure 6-24 on page 158. Here, you have to define the date and time for this process to run.

Note: Changes apply only when the process is in development mode.

4. If you use NetView SmartSets, as described in 6.4, “Preparing NetView for data collection” on page 129, you have to synchronize the hour when the SmartSet data is loaded to the NetView source database specified in the tdwdaemon.properties file and the schedule of the NetView warehouse pack ETLs.

   We recommend that the ETLs be scheduled at least one hour later then the SmartSets loading time. In our case study installation, the SmartSet data load is done at 11 p.m. as shown in Example 6-2 on page 137; one hour later at 0 hours, the ANM_c05_ETL1_Process is scheduled, as shown in Figure 6-24.

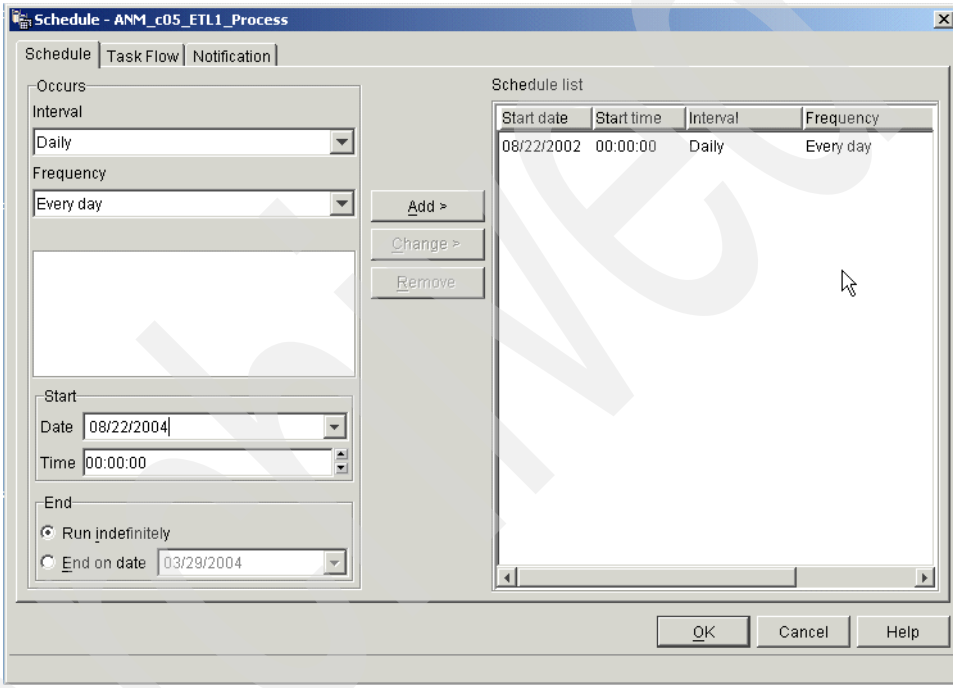


Figure 6-24 Schedule configuration for ANM_C05_ETL1_Process

6.6.4 Promoting the ETLs to Production mode

All of the processes are composed by components that have Development or Test status set as default. In order for them to run, their status needs to be changed from Development to Production mode. They are:

- ANM_c05_ETL1_Process:
  - ANM_c05_s010_extractNodeInfo
  - ANM_c05_s020_transformNodeInfo
  - ANM_c05_s030_loadNodeInfo

  - ANM_m05_s010_metric (This step is a link and cannot be promoted to Test, Development or Production)

- ANM_m05_ETL2_Process:
  - ANM_m05_s010_metric
  - ANM_m05_s020_fact
  - ANM_m05_s030_outage_rollUp
  - ANM_m05_s040_transition_rollup
  - ANM_m05_s050_ss_trans_rollup
  - ANM_m05_s060_out_rollup
  - ANM_m05_s070_total
- AN1_c05_SNMP_ETL1_Process:
  - AN1_c05_s010_extractsnmpdata
  - AN1_c05_s020_transformsnmpdata
  - AN1_c05_s030_loadsnmpdata

This step must be performed for all processes described above. Here, we use ANM_c05_ETL1_Process to describe it.

On the control server, using the Data Warehouse Center tool, select the previously listed processes and right-click them. Choose **Mode** → **Production**, as shown in Figure 6-25.

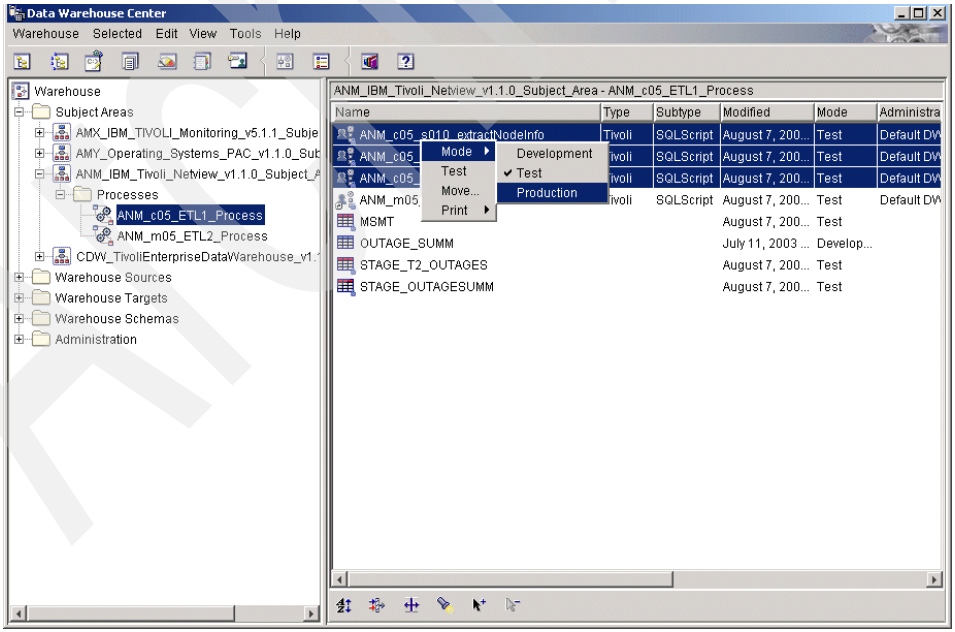


Figure 6-25 Promote ANM_c05_ETL1_Process

After promoting the processes to Production mode, they are scheduled for the configured times and they are visible in the Work in Progress list.

6.7 Reporting

In this section, we show how to set up, configure, and use some of the reports provided by the NetView availability warehouse pack (ANM).

Here is a list of the predefined reports provided by the IBM Tivoli NetView warehouse pack:

- Daily Status Summary By SmartSet
- Nodes With Longest Outage Time In Routers SmartSet
- Nodes With Most Status Changes In Routers SmartSet
- Nodes With The Longest Outage Times
- Nodes With The Most Daily Status Changes
- Summary Of Daily Network Status
- Summary Of Total Outage Time By SmartSet
- Summary Of Total Status Changes By SmartSet
- Total Daily Status Changes In Monitored Network
- Total Daily Status Changes In Routers SmartSet

Crystal Enterprise Professional Version 9 for Tivoli has, in comparison to a full Crystal license, reduced configuration options. If the reports shipped with IBM Tivoli NetView warehouse pack do not match your needs, and you want to develop additional reports, you have to upgrade your Crystal Enterprise installation.

Note: As described in Chapter 5, “Getting Tivoli Data Warehouse V1.3 up and running” on page 91, an ODBC connection to the data mart database needs to be defined on the Crystal Enterprise server before we can work with the reports. Refer to that chapter for details.

6.7.1 Accessing the Crystal ePortfolio feature

All reports provided by Tivoli Data Warehouse V1.3 must be accessed using the Crystal Enterprise ePortfolio feature. The ePortfolio can be accessed from the Crystal Enterprise Launchpad, as shown in Figure 6-26 on page 161:

```
http://<CRYSTALSERVER>/crystal/enterprise10/
```

In this URL, `<CRYSTALSERVER>` is the host name of the Crystal Enterprise Professional for Tivoli server machine.

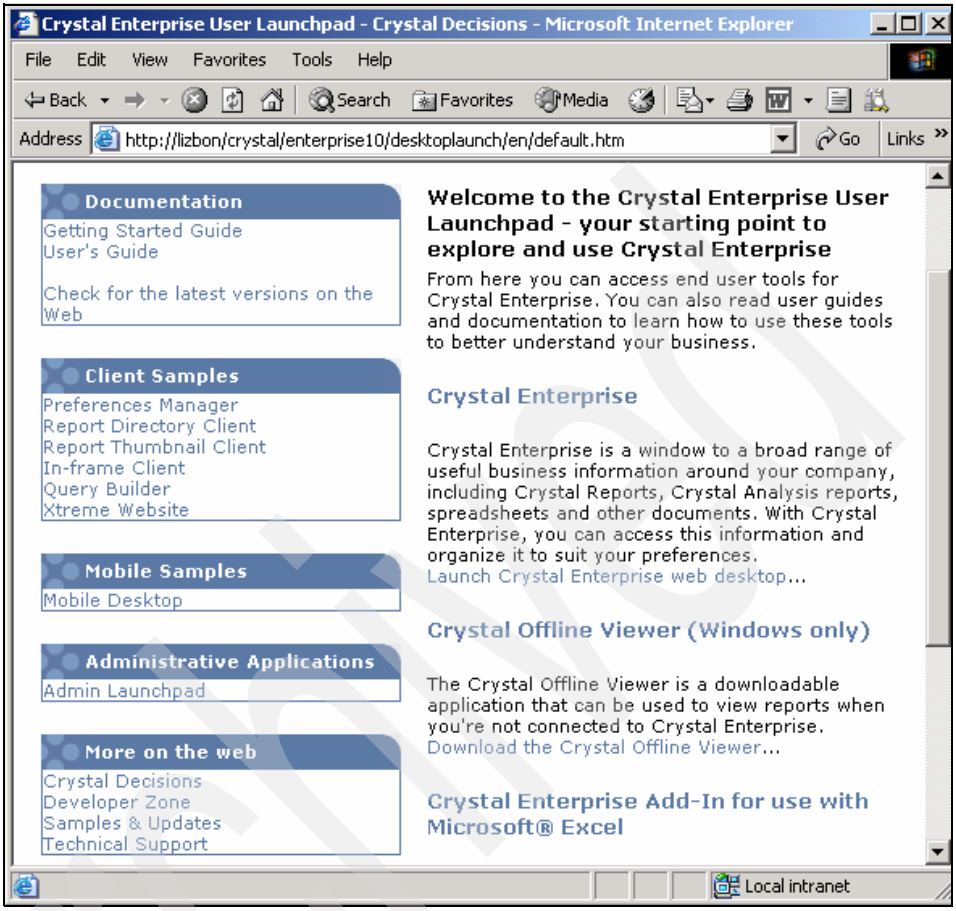


Figure 6-26 Crystal Enterprise: Launchpad

In this section, we concentrate on viewing NetView reports, and we do not explain the configuration of Crystal Enterprise to its full extent. For details about configuration and administration tasks, refer to the following guides shipped with the product:

- *Crystal Enterprise 10 Installation Guide*
- *Crystal Enterprise 10 Administrator's Guide*
- *Crystal Enterprise 10 Getting Started Guide*
- *Crystal Enterprise 10 User's Guide*

Note: Because we used a beta version of the Tivoli Data Warehouse V1.3 product in this IBM Redbook project, the window you will see in the general availability version of the product might be slightly different than ours.

From the Crystal Enterprise Launchpad, proceed by selecting the **Crystal Enterprise** link, which opens the window shown in Figure 6-27. In the top bar, you can see that we are authorized as user Guest. By default, the guest user has no access to the NetView reports, as indicated by the words *No folders* on the left side of the window.

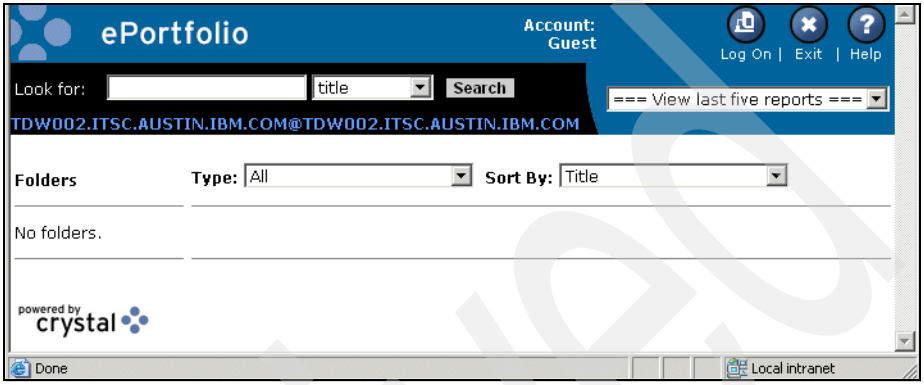


Figure 6-27 Crystal Enterprise: ePortfolio

The installation process of the first warehouse pack on the Tivoli Data Warehouse environment creates a user ID on the Crystal Enterprise environment named Tivoli. This user ID is to be used to access the reports provided by any IBM Tivoli software.

Archived

To log on as the Tivoli user ID, select the **Log On** button in the upper-right corner of the ePortfolio window, as shown in Figure 6-27 on page 162. The Log On window, as shown in Figure 6-28, opens. The Tivoli user ID has no password by default. We use the **Enterprise** authentication method, as we have specified during the Crystal Enterprise installation.

Figure 6-28 Crystal Enterprise: Log On

Archived

After entering the required data, select **Log On** to proceed. Now we are back at the ePortfolio window, as shown in Figure 6-29, but now with user Tivoli authority. Instead of No folders in the guest users ePortfolio window, there is now a link visible with the name *IBM Tivoli NetView* in the Tivoli user ePortfolio window, as shown in Figure 6-29.

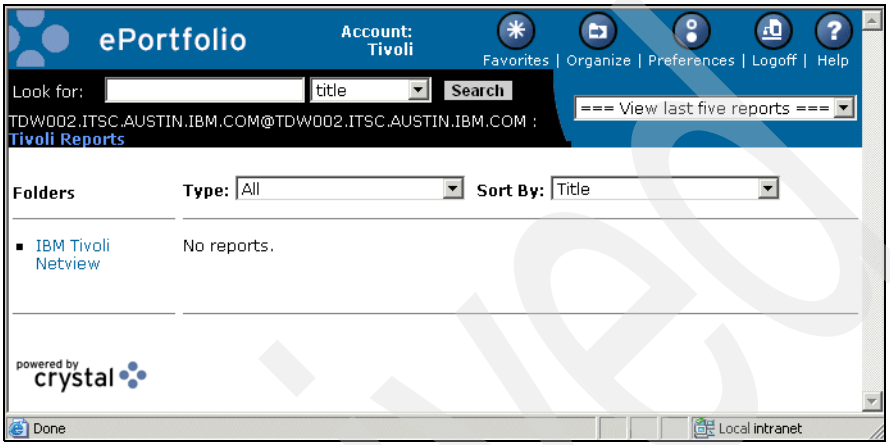


Figure 6-29 Crystal Enterprise: Folders

Archived

We follow this link by selecting **IBM Tivoli NetView** and proceed to the IBM Tivoli NetView reports, as shown in Figure 6-30. All reports provided by the IBM Tivoli NetView warehouse pack are listed there. As already mentioned, there are only reports on availability and no reports on performance.

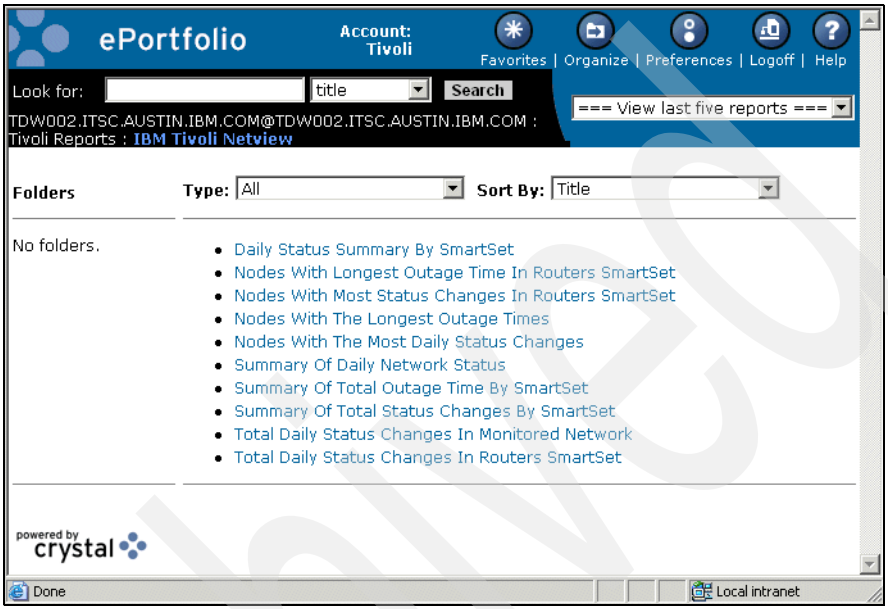


Figure 6-30 Crystal Enterprise: Tivoli Reports - IBM Tivoli NetView

We open the reports context menu by left-clicking the desired report name, as shown in Figure 6-31 on page 166. We are presented with a menu that contains the following items:

- View: Generate report instantaneously.
- View Latest Instance: View last report.
- Schedule: Change or create a new schedule for report generation.
- History: View already generated reports.

Archived

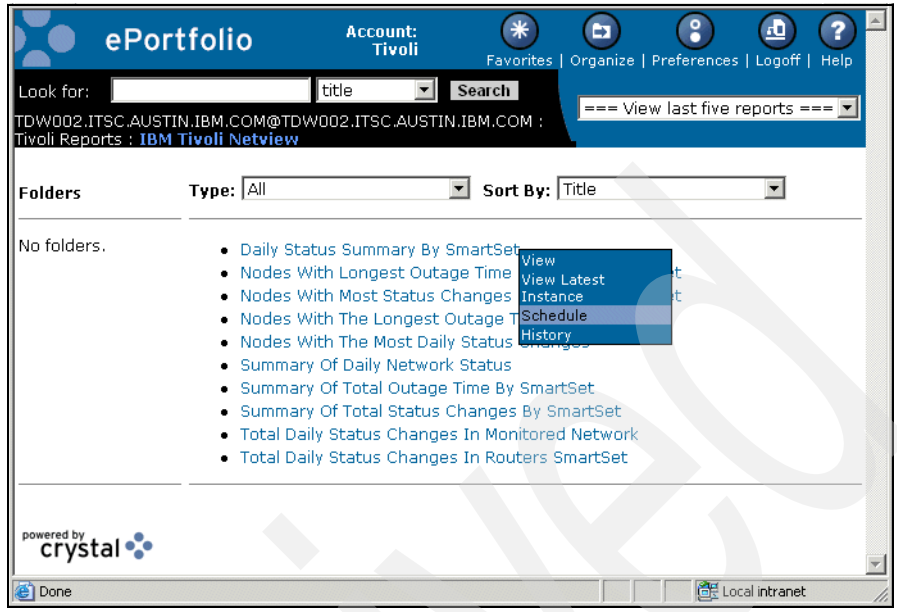


Figure 6-31 Crystal Enterprise: Daily Status Summary by SmartSet

We continue by selecting **Schedule** from the Daily Status Summary by SmartSet report, for example. The Schedule window, as shown in Figure 6-32, opens.

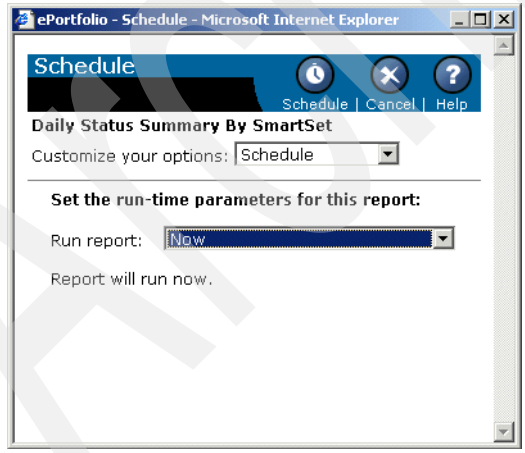


Figure 6-32 Crystal Enterprise: Schedule

The Customize your options toolbar offers three options:

- Schedule: Selecting this option starts a new schedule with the current options and parameters.

- Cancel: Selecting this option closes the Schedule window, and you return to the Reports window without adding a new schedule for the report.
- Help: Selecting this option opens the Help window.

Figure 6-33 shows the selection of parameters for the Schedule option. Here, you can select the frequency the reports should be generated.

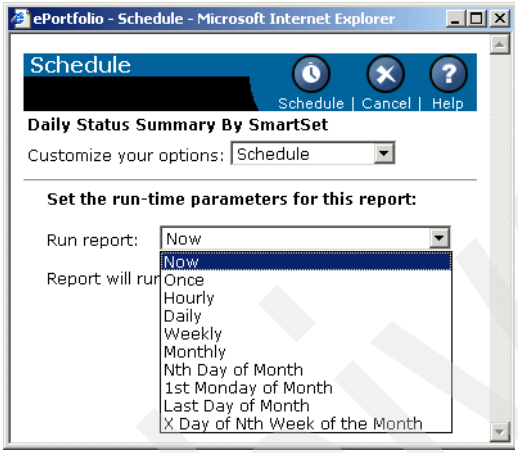


Figure 6-33 Crystal Enterprise: Parameters for schedule option

We want to schedule the report to run now.

Next, it is necessary to provide the required parameters of the report. From the Customize your options pull-down menu, select **Parameters**, as shown in Figure 6-34 on page 168. We left the other options settings to the default values.

Archived

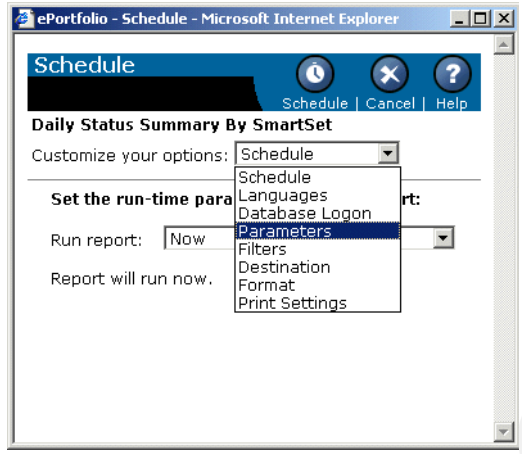


Figure 6-34 Crystal Enterprise: Schedule parameters

The Schedule windows changes to the window shown in Figure 6-35 on page 169, and we are presented with three selection fields:

- Time Filter
- General Time Frame
- Specific Time Frame

Note: Schedule requirements can differ for each report. The schedule selections presented here are for the Daily Status Summary by SmartSet report.

Archived

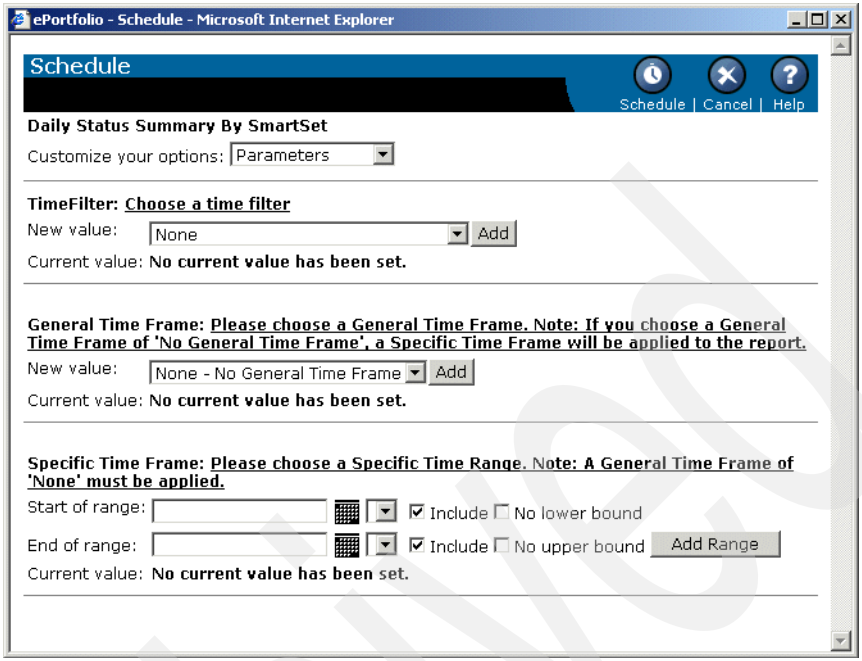


Figure 6-35 Crystal Enterprise: Schedule parameter selection

For the items Time Filter and General Time Frame, we select the default values **None** by clicking the **Add** button at each selection.

Therefore, we specify a lower and an upper bound for the specific time frame by selecting the Start of range and End of range parameters. Select **Add Range** to accept the settings. Figure 6-36 on page 170 shows the parameters window after the selections for our case study report.

Archived

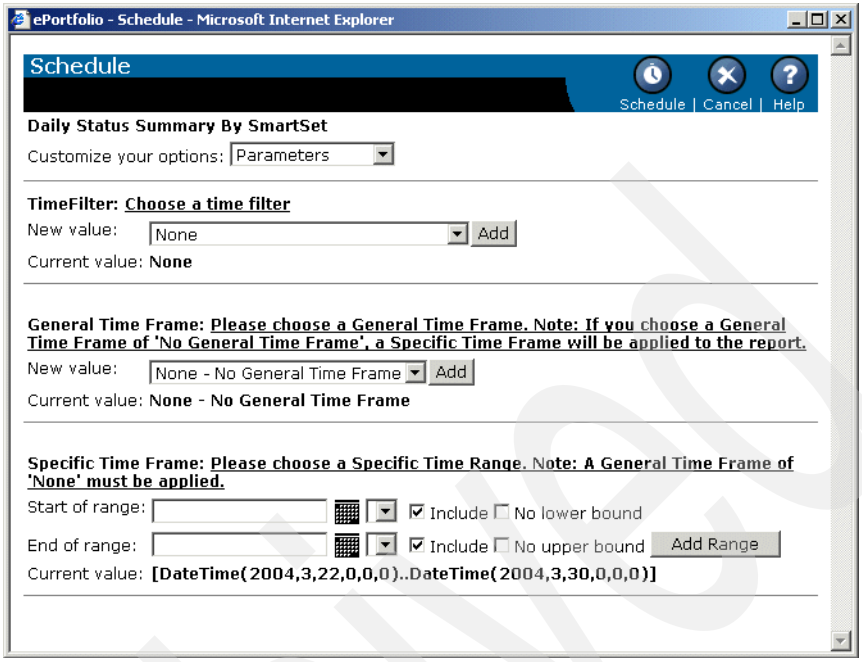


Figure 6-36 Crystal Enterprise: Parameters - Specific Time Frame

Now all required parameters are specified. Start the report generation by clicking the **Schedule** button from the toolbar.

Archived

As we have left the Schedule parameter set to **Now**, as shown in Figure 6-32 on page 166, the report is scheduled immediately, and the report's History window opens, as shown in Figure 5-37.

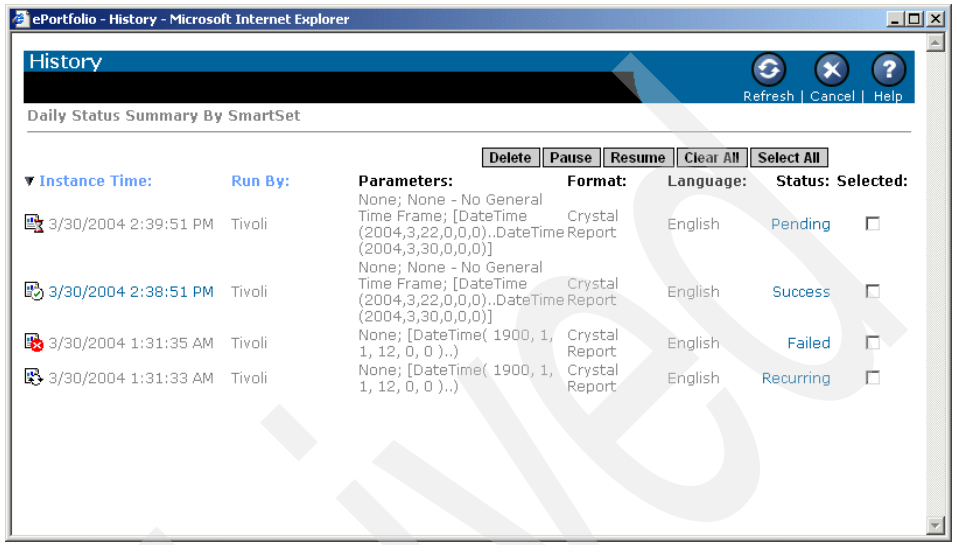


Figure 6-37 Crystal Enterprise: Report History

The report just scheduled is still running, and therefore, it is in status *Pending*.

> **Note:** The History window is not updated automatically. Click the **Refresh** button to view the current state.

Figure 6-37 shows four different status, as follows:

- Pending: Report generation is still running.
- Success: Report is generated successfully. Click the **Instance Time** link in the left column of the table to view the report.
- Recurring: Report is scheduled to be generated at certain times. Refer back to Figure 6-32 on page 166.

Archived

- Failed: Report generation failed. Click the **Failed** link to open the log window, as shown in Figure 6-38. The error message, `Information is needed before this report can be processed`, means that your parameter settings are not valid. Go back to the window shown in Figure 6-35 on page 169 and reenter your parameter settings.

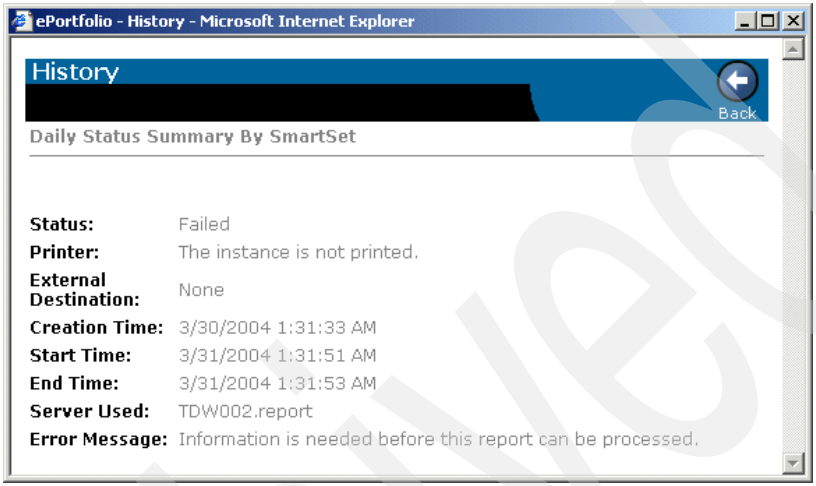
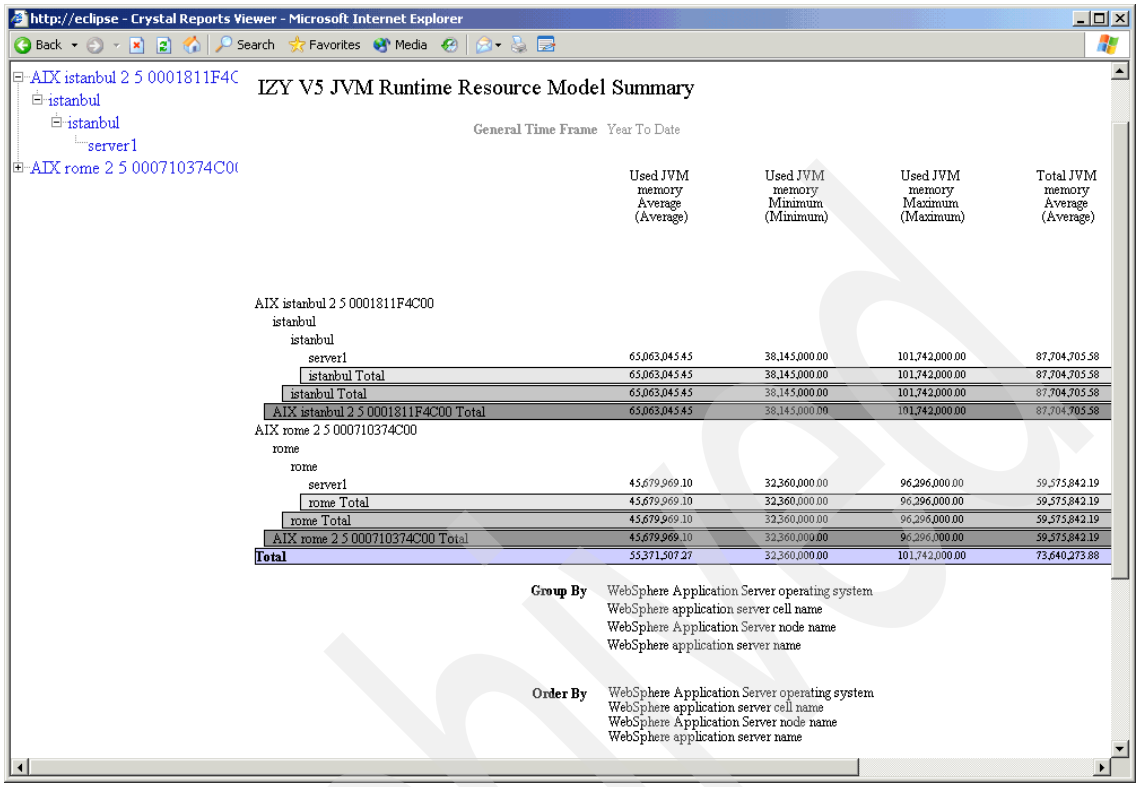


Figure 6-38 Failed report generation

6.7.2 Example reports

Figure 6-39 on page 173 shows the “Nodes With The Longest Overall Outage Times” report.

Archived



http://eclipse - Crystal Reports Viewer - Microsoft Internet Explorer

AIX istanbul 2 5 0001811F4C
istanbul
istanbul
server1
AIX rome 2 5 000710374C0

IZY V5 JVM Runtime Resource Model Summary

General Time Frame Year To Date

	Used JVM memory Average (Average)	Used JVM memory Minimum (Minimum)	Used JVM memory Maximum (Maximum)	Total JVM memory Average (Average)
AIX istanbul 2 5 0001811F4C00				
istanbul				
istanbul				
server1	65,063,045.45	38,145,000.00	101,742,000.00	87,704,705.58
istanbul Total	65,063,045.45	38,145,000.00	101,742,000.00	87,704,705.58
istanbul Total	65,063,045.45	38,145,000.00	101,742,000.00	87,704,705.58
AIX istanbul 2 5 0001811F4C00 Total	65,063,045.45	38,145,000.00	101,742,000.00	87,704,705.58
AIX rome 2 5 000710374C00				
rome				
rome				
server1	45,679,969.10	32,360,000.00	96,296,000.00	59,575,842.19
rome Total	45,679,969.10	32,360,000.00	96,296,000.00	59,575,842.19
rome Total	45,679,969.10	32,360,000.00	96,296,000.00	59,575,842.19
AIX rome 2 5 000710374C00 Total	45,679,969.10	32,360,000.00	96,296,000.00	59,575,842.19
Total	55,371,507.27	32,360,000.00	101,742,000.00	73,640,273.88

Group By WebSphere Application Server operating system
WebSphere application server cell name
WebSphere Application Server node name
WebSphere application server name

Order By WebSphere Application Server operating system
WebSphere application server cell name
WebSphere Application Server node name
WebSphere application server name

Figure 6-39 Nodes With The Longest Overall Outage Times report

Figure 6-30 on page 165 shows the “Nodes With The Most Daily Status Changes.” We see that lon-nt-db01 had 348 status changes within the defined time frame.

Archived

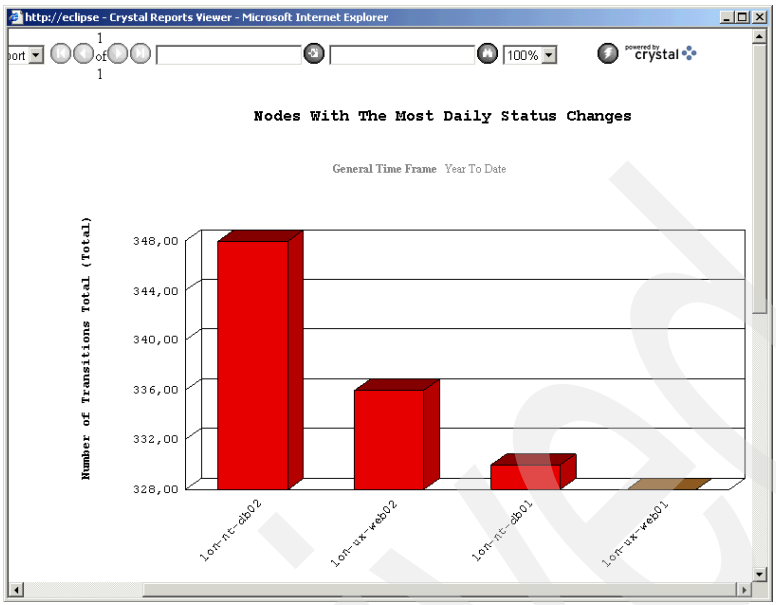


Figure 6-40 Report example

Archived

7

Performance maximization techniques

This chapter provides an in-depth look at the techniques you can use to improve the performance of your Tivoli Data Warehouse V1.3 environment. We cover the following topics:

- DB2 performance
- Operating system performance tuning
- Tivoli Data Warehouse performance

© Copyright IBM Corp. 2005. All rights reserved.

7.1 DB2 performance

DB2 is a big part of your Tivoli Data Warehouse V1.3 environment. Treating DB2 as a “black box” will have a huge impact on the overall performance of your Tivoli Data Warehouse V1.3 environment. Performance is vital for any data warehousing application. In order to get the most out of DB2, it needs to be carefully monitored and tuned.

The following topics introduce some of the factors affecting DB2 performance. This is based on the information provided by the IBM DB2 developerWorks® articles “Best practices for tuning DB2 UDB v8.1 and its databases: A handbook for high performance” and “Improve Database Performance on File System Containers in IBM DB2 Universal Database Stinger using Concurrent I/O on AIX” and the DB2 documentation. See “Related publications” on page 273 for information about these publications.

7.1.1 Introduction

Before we actually start talking about how to tune DB2, we provide you with an overview of some of the important factors affecting DB2 performance, such as buffer pools, indexes, and the hardware and operating system configuration.

We cover basic DB2 database administration topics, such as using the Configuration Advisor and the Design Advisor, as well as intermediate topics such as tablespace design and snapshot monitoring.

7.1.2 Buffer pools

Buffer pools improve database performance. A buffer pool is an area of memory into which database pages are read, modified, and held during processing.

If a needed page of data is already in the buffer pool, that page is accessed faster than if that page had to be read directly from disk. The database manager has agents whose tasks are to retrieve data pages from disk and place them in the buffer pool and to write modified data pages from the buffer pool back to disk. Retrieving these pages from disk is done by prefetchers; writing the modified pages back to disk is done by page cleaners.

With the current 32-bit operating systems, it is important to understand that there are limits on how much shared memory can be allocated. These limits restrict the database’s buffer pools (database global memory). However, 64-bit system do not have this limit.

The following list shows the operating systems and their shared memory limit:

- AIX: 1.75 GB
- Linux: 1.72 GB
- Sun: 3.35 GB
- Windows: 2-3 GB (use a 3 GB switch in boot.ini on Windows NT, 2000, and XP)

Use the following formula to calculate your approximate database global memory usage:

```
buffer pools + dbheap + util_heap_sz + pgkcachesz + aslheapsz + locklist +
approx. 10% overhead
```

If INTRA_PARALLEL is enabled, add the value of sheapthres_shr to the total.

How many buffer pools?

The choice of having one or more buffer pools depends on the amount of time you want to spend managing your DB2 environment.

If you do not want to spend your time fine-tuning DB2, choose one big buffer pool. DB2 is very good at self-tuning its buffer pools and keeping the most frequently accessed rows in memory, so one buffer pool might be all you need.

If you have the time, a test environment, and you are in need of performance improvements, you might want to use multiple buffer pools. The idea is to keep the most frequently accessed rows in a buffer pool. Sharing a buffer pool with tables that are accessed randomly or infrequently can cause “polluting” of your buffer pool by consuming space and possibly pushing a frequently accessed row to disk for a row that may never be used again. Keeping indexes in their own buffer pool can also significantly improve performance when indexes are heavily used.

For example:

- A medium-sized buffer pool for temporary tablespaces
- A large-size buffer pool for index tablespaces
- A large-sized buffer pool for tablespaces that contain frequently accessed tables
- A small-size buffer pool for tablespaces that contain infrequently accessed, randomly accessed, or sequentially accessed tables

Archived

How much memory?

A good starting point when sizing your buffer pools is to analyze your server's memory and ask yourself two questions:

1. How much physical memory does my server have?
2. Besides DB2, are there any other applications that run on my server?

Subtract the result of question number two from the result of question number one and you have the available memory. The operating system also needs memory to perform, so a guideline is to use 75-80% of the available memory for DB2.

For example:

- Server physical memory: 2 GB
- Memory usage other applications: 512 MB
- Available memory: 1.5 GB

Of this available memory, 75-80% (approximately 1 GB) can be used for DB2, leaving enough memory for the OS (approximately 500 MB) to consume. Remember that all of the database global memory has to be allocated from this 75-80%. You cannot use this for buffer pools only.

> **Important:** Be very careful not to allocate more memory than you have available or your performance will suffer from operating system memory paging.

Using block-based buffer pools

Complex queries, such as the queries used during the extract transform and load (ETL) phase in Tivoli Data Warehouse V1.3, might benefit from block-based buffer pools.

By default, the buffer pools are page-based, which means that contiguous pages on disk are prefetched into non-contiguous pages in memory. Sequential prefetching can be enhanced if contiguous pages can be read from disk into contiguous pages within a buffer pool. You can create block-based buffer pools for this purpose.

The NUMBLOCKPAGES parameter in the CREATE and ALTER BUFFERPOOL statement is used to define the size of the block memory. The BLOCKSIZE parameter specifies the size of the blocks and, therefore, the number of pages to be read from disk in a block I/O.

Archived

Deciding how much memory to dedicate to these blocks within the buffer pool is more complicated. A general rule is to set the BLOCKSIZE equal to the tablespace's EXTENT SIZE and to set the NUMBLOCKPAGES to 15% of the total buffer pool size. This can always be changed later, based on snapshot monitoring (see 7.1.6, "Snapshot monitoring" on page 197).

7.1.3 Tablespaces

DB2 has two types of tablespaces: system-managed space (SMS) tablespaces and database-managed space (DMS) tablespaces.

SMS tablespaces are managed by the operating system and, for the most part, allocate and deallocate space as needed. This type of tablespace uses a directory on the operating system to store database objects.

DMS tablespaces are managed by the database manager. The storage objects that are attached to a DMS tablespace can be files or raw devices. DMS tablespaces come with more management overhead, because you have to define the size of the containers.

A lot has been written about what tablespace type to use for your database. In general, the consensus seems to be:

- SMS is easy to manage
- DMS file is fast
- DMS raw is fastest but harder to manage

The recommendation for Tivoli Data Warehouse V1.3 is to use SMS for system temporary tablespaces and the system catalog tablespace. DMS file containers should be used for all other tablespaces.

> **Tip:** When using SMS, enable multipage file allocation by running the db2empfa tool. This can significantly improve performance, because it allows for the tablespace to grow one extent at the time rather than one page at the time.

Tablespace definitions

As with buffer pools, you need a tablespace for each page size being used. This means having a system temporary tablespace for each page size. You would then assign all tablespaces that share a matching page size to a buffer pool with the same page size.

Archived

The general recommendation when creating multiple tablespaces is to create for each page size a:

- System temporary tablespace
- Regular tablespace for indexes
- Regular tablespace for frequently accessed tables
- Regular tablespace for infrequently and randomly accessed tables

This is obviously very much dependent on the amount of available hard drives for your Tivoli Data Warehouse V1.3 environment.

Important: Before creating DMS file containers on Windows systems, be sure to defragment your disks.

Now, we provide several tablespace configuration recommendations on a system with 8, 10, and 12 hard drives.

In the eight disk scenario, the first thing we do is to separate the DB2 transaction logs from the data. The ETL1 inserts large amounts of data into the central data warehouse, which means log writing can become the cause for a lot of I/O contention. Second is to separate the central data warehouse indexes from the central data warehouse data to improve index performance and overall I/O parallelism.

For the data mart, we also placed the DB2 transaction logs on the same logging disk and spread the other tablespaces over the remaining two disks. Table 7-1 shows this eight disk scenario.

Table 7-1 Tivoli Data Warehouse eight disk setup

Disk number	Purpose
Disk 0	Operating system and DB2 binaries
Disk 1	DB2 transaction logs
Disk 2	Central data warehouse data
Disk 3	Central data warehouse data
Disk 4	Central data warehouse data
Disk 5	Central data warehouse indexes
Disk 6	Data mart data and indexes
Disk 7	Data mart data and indexes

In the next scenario, we have 10 hard drives for our environmental. The main difference with the eight disk scenario is that one drive will be added to the file system (striped) for the DB2 transaction logs to lower the disk utilization. The second new drive will be added to the data mart to be able to separate the indexes from the data. Table 7-2 shows the ten disk setup.

Table 7-2 Tivoli Data Warehouse 10 disk setup

Disk number	Purpose
Disk 0	Operating system and DB2 binaries
Disk 1	DB2 transaction logs
Disk 2	DB2 transaction logs
Disk 3	Central data warehouse data
Disk 4	Central data warehouse data
Disk 5	Central data warehouse data
Disk 6	Central data warehouse indexes
Disk 7	Data mart data
Disk 8	Data mart data
Disk 9	Data mart indexes

In the last scenario, we added another two drives to our environment. The differences with the 10 drive scenario are all central data warehouse related. We added one disk for central data warehouse data and one disk for central data warehouse temporary data.

Because of the large sorts during the ETL runs, there are bound to be sort overflows. These overflows will go to the temporary tablespace. Having a separate disk for temporary data will lower disk contention. The added disk for central data warehouse data will improve overall I/O parallelism. Another option would be to add both new disks for central data warehouse temporary data (striped). Table 7-3 shows the 12 disk setup.

Table 7-3 Tivoli Data Warehouse 12 disk setup

Disk number	Purpose
Disk 0	Operating system and DB2 binaries
Disk 1	DB2 transaction logs
Disk 2	DB2 transaction logs

Archived

Disk number	Purpose
Disk 3	Central data warehouse data
Disk 4	Central data warehouse data
Disk 5	Central data warehouse data
Disk 6	Central data warehouse data
Disk 7	Central data warehouse temp data
Disk 8	Central data warehouse indexes
Disk 9	Data mart data
Disk 10	Data mart data
Disk 11	Data mart indexes

Where to actually place the tablespace containers is harder to define. Ideally, you would want to have one tablespace container per disk to guarantee maximum I/O parallelism. In reality, a lot of customers use disk arrays for their production systems, which makes this solution very hard to implement.

Continuing with the assumption that the drives are part of a disk array, the recommendation would be to use a single container per tablespace. Additionally, you will need to set the DB2 profile registry variable DB2_PARALLEL_IO for that tablespace to enable I/O parallelism. More information about this variable can be found in 7.1.4, “Indexes and the Design Advisor” on page 185.

Important: These recommendations only apply to custom installations with predefined central data warehouse and data mart databases. For more information about predefined databases, refer to Chapter 14, “DB2 Data Warehouse essentials” in *Installing and Configuring Tivoli Data Warehouse*, GC32-0744.

Prefetching for performance

Setting the prefetch size correctly will improve the database manager’s ability to read pages in advance and reduce execution times. The default value is determined by the database configuration parameter DFT_PREFETCH_SZ. It is however, possible and recommended to change the prefetch size of your tablespaces.

The best solution is to set the prefetch size to a multiple of the extent size.

The optimal setting for prefetching on disk arrays is:

```
Prefetchsize = extent size * (# non-parity disks in array)
```

The optimal setting for prefetching on non-disk arrays is:

```
Prefetchsize = extent size * (# containers of the tablespace in different
physical disks)
```

To alter the prefetch size, use the following command:

```
alter tablespace <tablespace name> prefetch size <value>
```

Enabling concurrent I/O

The following is based on the IBM DB2 developerWorks article “Improve database performance on file system containers in IBM DB2 UDB V8.2 using Concurrent I/O on AIX,” available at:

http://www.ibm.com/developerworks/db2/library/techarticle/dm-0408lee/index.html

Before DB2 Version 8.2 Fix Pack 4 (on AIX), a file read request required the file system to first read from disk into the system cache and into the DB2 internal buffer pool and then copy the data to the user's buffer. For a file write request, the data is copied from the user's buffer into both caches and eventually back to disks. This dual level of cache is not necessary. Direct I/O is an alternative caching policy that bypasses the default file system buffer caching. The benefits of using direct I/O includes reducing CPU utilization for file reads and writes by eliminating the copy from the cache to the user buffer. It also avoids diluting the effectiveness of caching of other files in the case of a poor cache hit ratio.

DB2 V8.2 Fix Pack 4 introduced limited direct I/O support for AIX (profile registry variable DB2_DIRECT_IO). This support is for all SMS containers except for long files, LOBs, and temporary tablespaces. (On Windows, all SMS and DMS containers are supported by the profile registry variable DB2NTNOCACHE.)

One of the new features in DB2 V8.2 is support for AIX concurrent I/O. The main difference between concurrent I/O and direct I/O is that direct I/O requires per-file write locks (inode locks) to ensure data integrity and improve fault tolerance. Unfortunately, this poses bottlenecks for database applications. Concurrent I/O does not need these write locks.

Under concurrent I/O, the inode lock is acquired in read-shared mode for both read and write access. However, in situations where the contents of the inode might change for reasons other than a change to the contents of the file, the inode lock is acquired in write-exclusive mode. A situation such as this occurs when a file is extended or truncated. Both these actions require updates to the “table of contents” of the corresponding inode.

Requirements

To be able to use direct I/O or concurrent I/O in your Tivoli Data Warehouse V1.3 environment, the requirements in Table 7-4 have to be met.

Table 7-4 Concurrent I/O and direct I/O prerequisites

Platforms	File system types	Recommended fixes	Supported I/O mechanism
AIX 4.3.3+	JFS	None	Direct I/O
AIX 5L V5.1+	JFS, JFS2	None	Direct I/O
AIX 5L V5.2+	JFS	None	Direct I/O
AIX 5L V5.2+	JFS2	Maintenance level 3 and additional AIX APARs[a]	Concurrent I/O

a. Refer to the "Known issues for DB2 Universal Database on AIX 4.3.3, 5.1, 5.2, and 5.3" Web page for a list of specific APARs required, available at: http://www.ibm.com/support/docview.wss?rs=71&uid=swg21165448

Enablement at tablespace level

The new keywords NO FILE SYSTEM CACHING and FILE SYSTEM CACHING have been introduced to the CREATE and ALTER TABLESPACE SQL statements to enable the user to specify whether direct I/O or concurrent I/O is to be used for each tablespace.

By default, this new tablespace will be using buffered I/O; the FILE SYSTEM CACHING keyword is implied:

```
CREATE TABLESPACE <tablespace name>
```

The new keyword NO FILE SYSTEM CACHING indicates that caching at the file system level will be OFF for this particular tablespace:

```
CREATE TABLESPACE <tablespace name> NO FILE SYSTEM CACHING
```

The new keyword NO FILE SYSTEM CACHING indicates that caching at the file system level will be OFF for this particular tablespace

```
ALTER TABLESPACE <tablespace name> NO FILE SYSTEM CACHING
```

This method of enabling direct I/O or concurrent I/O gives the users control of the I/O mode at the tablespace level. It can be used for both SMS and DMS tablespaces except for the following:

- SMS large files
- SMS large object file
- SMS and DMS temporary tablespaces

The NO FILE SYSTEM CACHING keywords are ignored for the above cases.

Enablement at file system level

An alternate method for enabling direct I/O or concurrent I/O is to use special mount options when mounting a file system.

To enable direct I/O on JFS:

```
mount -o dio <file system name>
```

To enable concurrent I/O on JFS2:

```
mount -o cio < file system name>
```

The advantage of this method is that a complete file system can be enabled for concurrent I/O using only one command. All tablespace containers in that file system will be automatically enabled for concurrent I/O. It also means none of your existing tablespace creation scripts need to be modified.

7.1.4 Indexes and the Design Advisor

An index is a list of the locations of rows in a table, sorted by the contents of one or more specified columns. Indexes are typically used to speed up access to a table. However, they can also serve a logical data design purpose. For example, a unique index does not allow entry of duplicate values in the columns, thereby guaranteeing that no two rows of a table are the same. Both these types of indexes are used in Tivoli Data Warehouse V1.3.

The DB2 Design Advisor is used to recommend and evaluate indexes for specific SQL workloads. If you have a specific problem with ETLs in your Tivoli Data Warehouse V1.3 environment, you can copy the query (or queries) into the Design Advisor to have it recommend a set of efficient indexes.

In the following section, we provide a short case study about how the Design Advisor can help you speed up your ETLs in the Tivoli Data Warehouse V1.3 environment.

General case information

We will be running the IBM Tivoli Monitoring ETL1, also known as AMX. One million (1,000,000) rows will be transferred from the ITM_DB database into the TWH_CDW database. This ETL breaks down into five different steps, as shown in the right frame of the DB2 Data Warehouse Center, as shown in Figure 7-1 on page 186.

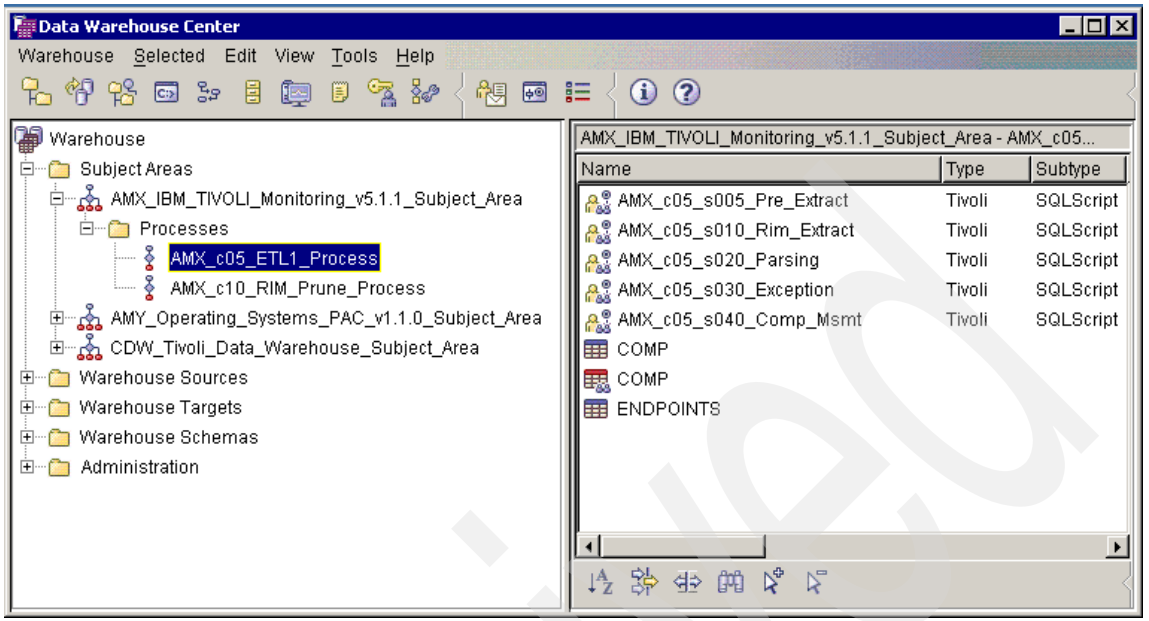


Figure 7-1 DB2 Data Warehouse Center

We focus on the last step AMX_c05_s040_Comp_Msmt, because this is the step that usually consumes most of the ETL runtime.

Analyzing ETL log files

By analyzing the DB2 Data Warehouse log files, we can find the runtimes for the individual SQL statements. The log file is in the %DB2HOME%\LOGGING or %VWS_LOGGING% directory on Windows.

The file we are looking for is amx_c05_s040_comp_msmt.log.

In this file, you will find many sections like the one displayed in Example 7-1.

Example 7-1 amx_c05_s040_comp_msmt log file

```
========================
= Script file line #  : 29
= Exec at Source DS   : TWH_CDW (IBM DB2)
= SQL Statement       : "insert into AMX.Stage_Comp_Level_Old select Comptyp_cd
as Comptyp_cd, msrc_cd as msrc_cd, lvl as lvl from AMX.Stage_Comp_Level"
= Elapsed Time        : 00:00:00.016
= Rows Modified       : 27
= Successful Execution: No Errors
========================
```

We are initially interested in the elapsed time. The format for the elapsed time is hh:mm:ss:zzz. In this particular case, the elapsed time was 16 milliseconds. We are looking for SQL statements that took more than 5 minutes to complete (everything above 00:05:00:00).

The statement shown in Example 7-2 will be used to feed the Design Advisor.

Example 7-2 Statements used to feed the Design Advisor

```
insert into AMX.ITM_Keys select a5.msrc_cd as PAC, a1.key_typ as property,
a1.key_val as property_val, a1.iid as iid, a1.eid as eid, a1.rcid as rcid,
a3.comptyp_cd as Comptyp_cd from AMX.Stage_Key_Parsed a1, AMX.Stage_Resources
a2, AMX.Resource_Transl a3, AMX.Stage_RmProfiles a4, AMX.Category_Transl a5
where a2.rcid = a1.rcid and a2.eid = a1.eid and a2.resource = a3.resource and
substr(a3.comptyp_cd,1,3)=a5.msrc_cd and a4.id=a2.id and a4.eid=a2.eid and
a4.category=a5.category group by a5.msrc_cd,a1.key_typ,a1.key_val, a1.iid,
a1.eid, a1.rcid, a3.comptyp_cd;
```

Now that we have found an SQL statement that takes more than 5 minutes to complete, highlight the SQL statement text and press Ctrl+C to copy the text to the Clipboard. Now, we start the Design Advisor.

Note: The DB2 Administration Server is needed to run the Design Advisor. Make sure that it is running on the system where the database you are connecting to resides (local or remote).

To start and work with the Design Advisor, complete the following steps:

1. First start the DB2 Control Center by selecting **Start** → **IBM DB2** → **General Administration Tools** → **Control Center**.
2. In the DB2 Control Center window, select **All Databases** to display the database list.
3. Right-click the **TWH_CDW** database icon and select **Design Advisor**, as shown in Figure 7-2 on page 188.

Archived

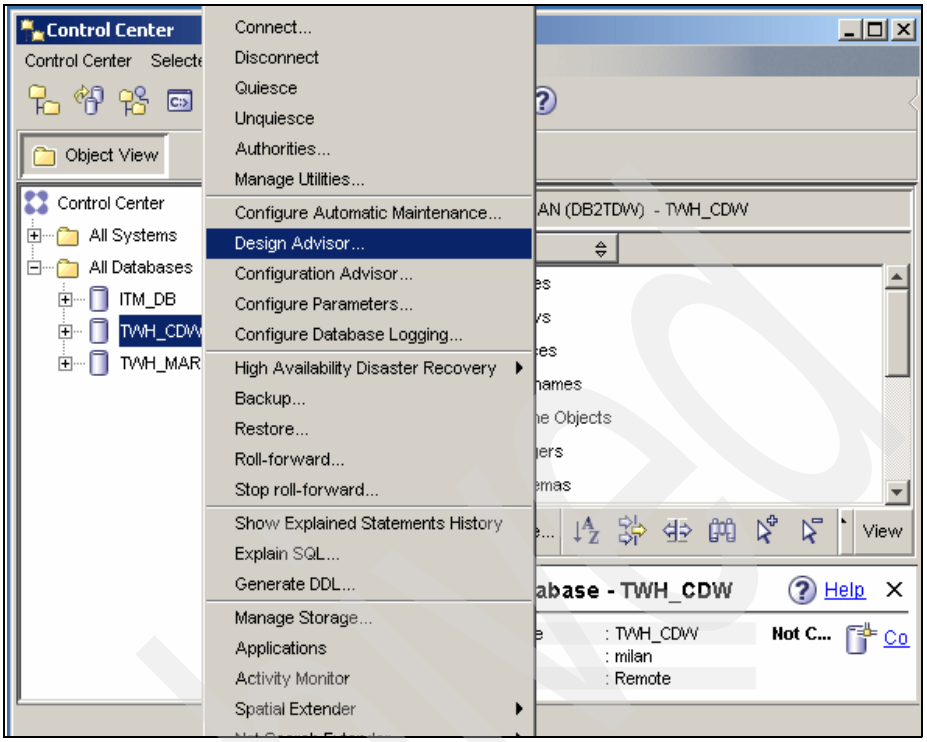


Figure 7-2 DB2 Control Center

4. When the Design Advisor window opens, click **Next** to go to the Features page.
5. Clear the **Materialized query tables (MQTs)** and **Multidimensional clustering (MDC) tables** options. We will concentrate on indexes for this workload.
6. Click **Next** to go to the Workload page. This is the most important page in the Design Advisor. This is where you “feed” the Design Advisor its workload.
7. We add the SELECT statement from the ETL log file to the workload. To do this, follow these steps:

   a. Type a name in the **Workload name** field.
   b. Choose **User ID** (or any other schema) from the Schema drop-down list.
   c. Click **Add** to open the Add SQL Statement window.
   d. Paste the SQL statement from the Clipboard into the text box.
   e. Click **OK** when you are done.

   The workload has now been added to the Design Advisor. It is possible to add more than one query to this workload. Follow steps c through e to add more SQL statements.

8. To continue with the Design Advisor, click **Next**. The Statistics page is the next important page in the Design Advisor. The DB2 optimizer uses statistics to determine the optimal path to the data. Refer to 7.1.7, “Maintenance” on page 208 for a detailed description of table statistics and how to keep them up to date.

   It is important that we update the statistics for the workload we just created. If your TWH_CDW database is not that big, you can select all tables by clicking the **>>** button. If your database is big, or you are not sure about the size, select all the tables the SQL query references and click the **>** button.

9. Click **Next** to go to the Options page. Leave everything on this page at its default value and click **Next**.

10. The Calculation page is the last page before the workload will be run. Clear the **Set maximum time to calculate recommendations** option. This will allow the Design Advisor to run until the statement has finished and possibly give you better recommendations. Click **Next** to run the workload.

11. After the workload has finished the Recommendations page is displayed. Figure 7-3 on page 190 shows the Recommendations page with two recommended indexes. One index already exists, the second one is new. This new index will give this workload an estimated 79% performance increase with an estimated disk usage of 1.2 MB.

Archived

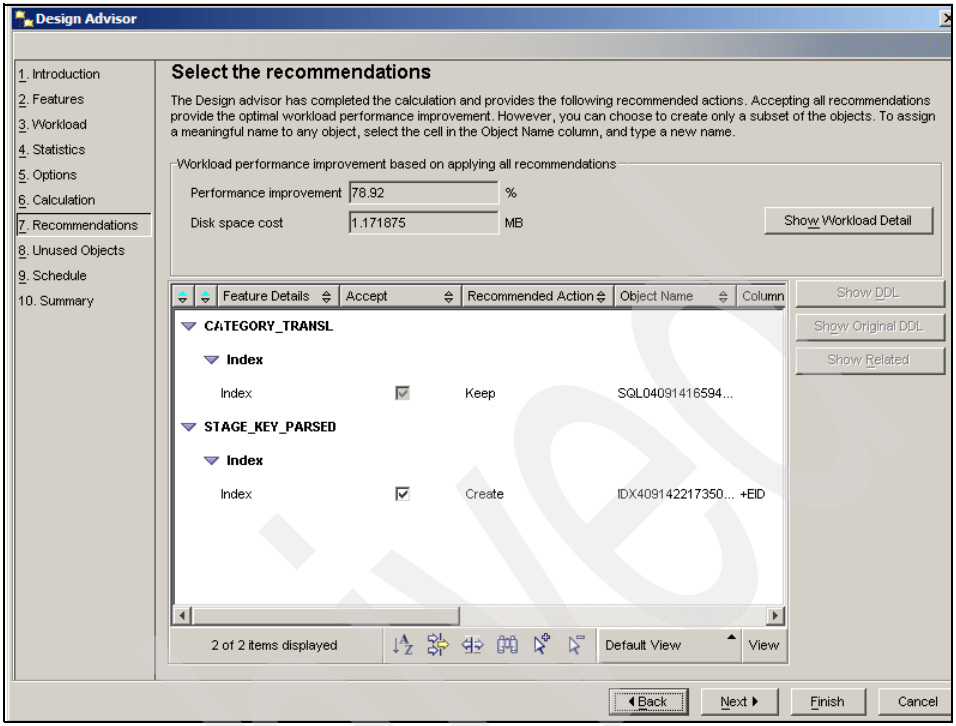


Figure 7-3 Design Advisor recommended indexes

12. Click **Next** to go to the Unused Objects page. Objects, mostly indexes, that were not used for this workload are displayed here. Do not select anything. Click **Next** to go to the Schedule page.
13. Select **Run now without saving task history** to add the recommended indexes directly, or select **Enable scheduler** to add these recommendations to the scheduler and run them later. Click **Next** to go to the Summary page.
14. Click **Finish** to run your choice from the Schedule page.

For users without access to a GUI, there is also a command line version of the Design Advisor. Example 7-3 shows you how to find the correct syntax.

Example 7-3 The db2advis syntax

```
C:\SQLLIB\BIN>db2advis -h

 db2advis is a tool that recommends database objects based
 on an existing database and a set of queries.

  SYNTAX:
```

```
db2advis -d <db name> [-t <time>] [-l <disk_space>]

         -s "sql stmt" | -i <infile> | -w <workload name> | -g | -qp
        [ -[user|a] <username> [/<password>] ]
        [ -[type|m] <advis_type> ]
        [ -[compress|k] <compression level> ]
        [ -[tbsp|b] <MQT creation table space> ]
        [ -[simtbsp|c] <simulation table space> ]
        [ -[schema|n] <schema name> ]
        [ -[qualifier|q] <qualifying schema name>]
        [ -[script|o] <output script> ]
        [ -[password|x] ] [ -[sample|r] ] [ -[deferred|u] ] [ -[drop|f] ]
```

A typical command would be:

```
db2advis -d sample -t 0 -i inputfile.sql -o outputfile.out
```

7.1.5 Environment settings and the Configuration Advisor

There are three main configuration areas for your DB2 environment.

- Profile registry: This area controls the way your DB2 engine behaves.
- Database manager configuration: This area controls the way your instance behaves. There is one configuration per instance.
- Database configuration: This area controls the way your database behaves. There is one configuration per database.

The profile registry can only be updated on the DB2 command line using the **db2set** command. The database and database manager configuration can be configured using the DB2 Control Center or using the DB2 command line tools.

We discuss all three in the following sections.

Configuration Advisor

The Configuration Advisor is a GUI tool that enables you to automatically configure the database manager and database configuration parameters based on the answers you provide to a series of questions. It is an excellent starting point for less experienced DB2 administrators to tune their Tivoli Data Warehouse V1.3 environment.

To launch and work with the Configuration Advisor, complete the following steps:

1. Select **Start** → **IBM DB2** → **General Administration Tools** → **Control Center**.
2. In the DB2 Control Center window, select **All Databases** to display the database list.

Archived

3. Right-click a database icon (for example, **TWH_CDW)** and select **Configuration Advisor**.
4. Follow the instructions provided in the application to answer the questions.

After you have answered all the questions, you can generate and apply the results. Instead of applying the results directly, you can also store them in the DB2 Task Center for later execution.

Figure 7-4 shows a sample Results page.

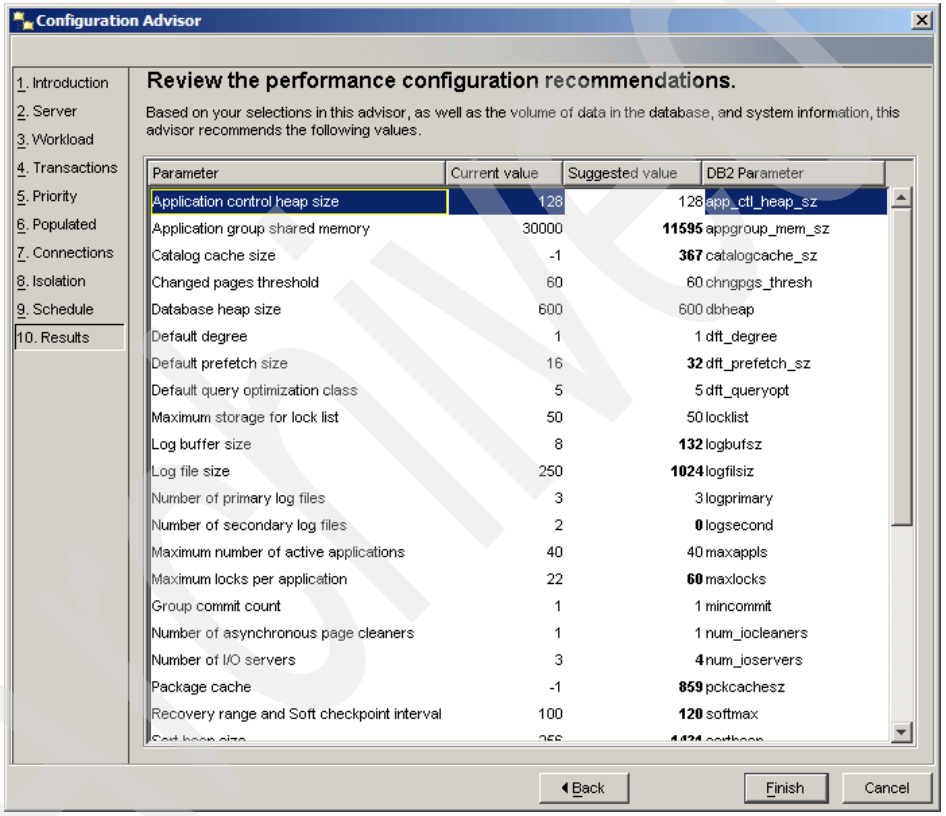


Figure 7-4 Configuration Advisor Results page

For users without access to a GUI, there is also a command line version of the Configuration Advisor called **`autoconfigure`**. Example 7-4 on page 193 shows you how to find the correct syntax and an example command.

Example 7-4 Autoconfigure command

```
milan:/home/db2tdw#  db2 ? autoconfigure
AUTOCONFIGURE [USING config-keyword value [{,config-keyword value}...]]
[APPLY {DB ONLY | DB AND DBM | NONE}]

config-keyword:
  MEM_PERCENT, WORKLOAD_TYPE, NUM_STMTS, TPM, ADMIN_PRIORITY, IS_POPULATED
  NUM_LOCAL_APPS, NUM_REMOTE_APPS, ISOLATION, BP_RESIZEABLE.

NOTE: From the operating system prompt, prefix commands with 'db2'.
      Special characters MAY require an escape sequence (\), for example:
      db2 \? change database
      db2 ? change database xxx comment with \"text\"

C:\SQLLIB\BIN>db2 autoconfigure using mem_percent 50 workload_type complex
is_populated yes apply db and dbm
```

The example **autoconfigure** command “feeds” the DB2 engine three values (mem_percent, workload_type, and is_populated) on which the engine can base its recommendations for the new database and database manager configuration values. The **autoconfigure** command requires a database connection.

The last part of the command, **apply db and dbm**, tells the **autoconfigure** command that it can apply the recommended values immediately. If you want to see the recommended changes but do not want to have them applied immediately, you can specify **apply none**.

Profile registry configuration

The DB2 profile registry variables typically affect how the optimizer and the DB2 engine behave. There is an enormous amount of registry variables, most of which will not be used.

To show a list of all available registry variables, type:

```
db2set -lr
```

To show a list of your currently set registry variables, type:

```
db2set -all
```

Example 7-5 on page 194 shows the output of this command.

Archived

Example 7-5 Output of the db2set command

```
C:\SQLLIB>db2set -all
[e] DB2PATH=C:\SQLLIB
[i] DB2ACCOUNTNAME=TDW1302\db2admin
[i] DB2INSTOWNER=TDW1302
[i] DB2PORTRANGE=60000:60003
[i] DB2_GRP_LOOKUP=LOCAL
[i] DB2INSTPROF=C:\SQLLIB
[i] DB2COMM=TCPIP
[g] DB2SYSTEM=TDW1302
[g] DB2PATH=C:\SQLLIB
[g] DB2INSTDEF=DB2
[g] DB2ADMINSERVER=DB2DAS00

C:\SQLLIB>
```

In Example 7-5, `[i]` means that this variable is set for the current instance, and `[g]` means that the variable is set globally (for all instances).

To set a registry variable at the instance level, type:

```
db2set -i variable=value
```

To set a registry variable at the global level, type:

```
db2set -g variable=value
```

For example, the following command sets the DB2_DIRECT_IO variable globally:

```
db2set -g DB2_DIRECT_IO=ON
```

Note: Do not place spaces between the variable and the value, or the variable will be reset to its default value.

The following list provides descriptions of registry variables that can be useful for your Tivoli Data Warehouse V1.3 environment:

- DB2_PARALLEL_IO: Enabling this variable ensures that even on single-container tablespaces, parallel I/O for individual prefetch requests can be done (which is useful when the single containers consist of more than one physical disk due to RAID or file system striping).

  To enable parallel I/O for all tablespaces, type:

  ```
  db2set -g DB2_PARALLEL_IO=*
  ```

  To enable parallel I/O for tablespaces with ID 3 and ID 4, type:

  ```
  db2set -g DB2_PARALLEL_IO=3,4
  ```


Archived

To enable parallel I/O for all tablespaces, telling DB2 that the tablespace containers are spread across four physical disks, type:

```
db2set -g DB2_PARALLEL_IO=*:4
```

To enable parallel I/O for with ID 3 and ID 4, telling DB2 that the tablespace containers for tablespace ID 3 and 4 are spread across four physical disks, type:

```
db2set -g DB2_PARALLEL_IO=3,4:4
```

- DB2_DIRECT_IO: Enabling this variable will eliminate file system caching for DB2 files on AIX. Refer to 7.1.3, “Tablespaces” on page 179 for a full description of direct I/O and concurrent I/O on AIX. To enable this parameter, type:

  ```
  db2set -g DB2_DIRECT_IO=ON
  ```

- DB2NTNOCACHE: Enabling this variable will eliminate file system caching for DB2 files on Windows. To enable this parameter, type:

  ```
  db2set -g DB2NTNOCACHE=ON
  ```

- DB2_USE_ALTERNATE_PAGE_CLEANING: When this variable is set to ON, DB2 uses a proactive method of page cleaning, writing changed pages to disk and proactively finding victims. Doing this allows the page cleaners to better use available disk I/O bandwidth. When this variable is set to ON, the chngpgs_thresh database configuration parameter is no longer relevant because it does not control page cleaner activity. To enable this parameter, type:

  ```
  db2set -g DB2_USE_ALTERNATE_PAGE_CLEANING=ON
  ```

- DB2_ANTIJOIN: Enabling this variable ensures that the optimizer searches for opportunities to transform NOT EXISTS subqueries into anti-joins, which can be processed more efficiently by DB2. To enable this parameter, type:

  ```
  db2set -g DB2_ANTIJOIN=YES
  ```

Note: Some registry variable defaults changed from DB2 V7.2 to DB2 V8.2 or became obsolete. These are not listed.

Database and database manager configuration

Previously, we had the Configuration Advisor change the database manager and database configuration parameters for us. In this section, we show how to change these manually using the DB2 command line. This is important for the snapshot monitoring discussed in 7.1.6, “Snapshot monitoring” on page 197.

Some of the database manager (DBM) configuration parameters can be changed online (take effect immediately), while others require the instance to be stopped and restarted. The same applies to the database configuration

parameters: Some take effect immediately, while others require the database to be deactivated and reactivated.

Viewing database manager and database configuration

To see the database manager configuration type, use the **get dbm cfg** command. For an overview of the database parameters, use the **get db cfg for <database name>** command.

Example 7-6 shows how to view the configurations for the database manager and the TWH_CDW database.

Example 7-6 Viewing database manager and database configuration

```
milan:/home/db2tdw#  db2 get dbm cfg

          Database Manager Configuration

     Node type = Enterprise Server Edition with local and remote clients

 Database manager configuration release level            = 0x0a00

 CPU speed (millisec/instruction)             (CPUSPEED) = 1.137562e-06
 Communications bandwidth (MB/sec)      (COMM_BANDWIDTH) = 1.000000e+02

 Max number of concurrently active databases     (NUMDB) = 2
 Data Links support                         (DATALINKS) = NO
 Federated Database System Support          (FEDERATED) = NO
 Transaction processor monitor name        (TP_MON_NAME) =

milan:/home/db2tdw#  db2 get db cfg for twh_cdw

      Database Configuration for Database twh_cdw

 Database configuration release level                    = 0x0a00
 Database release level                                  = 0x0a00

 Database territory                                      = US
 Database code page                                      = 1208
 Database code set                                       = UTF-8
 Database country/region code                            = 1
```

Updating the database manager and database configuration

To update the database manager configuration, use the **update dbm cfg using <parameter> <value>** command. To update the database configuration, use the **update db cfg for <database name> using <parameter> <value>** command.

Example 7-7 shows how to update a database manager and database configuration parameter.

Example 7-7 Updating database manager and database configuration

```
milan:/home/db2tdw#  db2 update dbm cfg using SHEAPTHRES 50000
DB20000I  The UPDATE DATABASE MANAGER CONFIGURATION command completed
successfully.
SQL1362W  One or more of the parameters submitted for immediate modification
were not changed dynamically. Client changes will not be effective until the
next time the application is started or the TERMINATE command has been issued.
Server changes will not be effective until the next DB2START command.

milan:/home/db2tdw#  db2 update db cfg for twh_cdw using dbheap 5000
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
```

As you can see in Example 7-7, the database manager configuration parameter was not updated dynamically. Another way of identifying whether a parameter is updated dynamically or not is by using the **get dbm cfg show detail** command. This will show you a current value and a delayed value. If the delayed value is different from the current value, you need to stop and start the instance.

The same can be done for delayed database configuration parameters. Use the **get db cfg for <database name> show detail** command to display the current and delayed values.

In the next two sections, we discuss the database manager snapshot, the database snapshot, and when to take action.

7.1.6 Snapshot monitoring

Now that you have used the Configuration Advisor to tune your database and database manager environment, and you have used the Design Advisor to add some indexes, it is time to see whether those changes made a difference or not.

Snapshot monitoring can help you identify the behavior of a database and the database manager over a period of time. It will show things such as memory utilization and locking. It is an excellent method to identify problems in your database environment.

There are six database areas that can be monitored:

- Buffer pool activity
- Locks
- Sorts
- Dynamic SQL statements
- Table activity
- Units of work

Besides these six database areas, it is also possible to monitor the database manager.

The **get monitor switches** command can be used to display your current monitor settings. Example 7-8 shows the output of the command.

Example 7-8 Monitor switches

```
milan:/cdw_data#  db2 get monitor switches

            Monitor Recording Switches

Switch list for db partition number 0
Buffer Pool Activity Information  (BUFFERPOOL) = OFF
Lock Information                        (LOCK) = OFF
Sorting Information                     (SORT) = OFF
SQL Statement Information          (STATEMENT) = OFF
Table Activity Information             (TABLE) = OFF
Take Timestamp Information         (TIMESTAMP) = ON  09-14-2004 11:15:33.146198
Unit of Work Information                 (UOW) = OFF
```

As you can see in the example, all monitors are switched off by default. The TIMESTAMP switch can be ignored. It is only used to add time stamp information to your snapshots.

To update the monitor switches, use the **update monitor switches** command, as shown in Example 7-9.

Example 7-9 Updating monitor switches

```
milan:/cdw_data#  db2 update monitor switches using lock on bufferpool on
DB20000I  The UPDATE MONITOR SWITCHES command completed successfully.

milan:/cdw_data#  db2 update monitor switches using table on sort on
DB20000I  The UPDATE MONITOR SWITCHES command completed successfully.

milan:/cdw_data#  db2 get monitor switches

            Monitor Recording Switches

Switch list for db partition number 0
Buffer Pool Activity Information  (BUFFERPOOL) = ON  09-15-2004 11:57:25.845431
Lock Information                        (LOCK) = ON  09-15-2004 11:16:15.282893
Sorting Information                     (SORT) = ON  09-15-2004 11:57:25.845431
SQL Statement Information          (STATEMENT) = OFF
Table Activity Information             (TABLE) = ON  09-15-2004 11:57:35.527302
Take Timestamp Information         (TIMESTAMP) = ON  09-14-2004 11:15:33.146198
Unit of Work Information                 (UOW) = OFF
```

Archived

Note: After updating the switches, the monitors are switched ON for the current session only. As soon as you use the **terminate** command or exit the session, the monitor switches will be reset to their initial status.

To prevent errors during monitoring, we recommend that you increase the size of the monitor heap to 1024. To do this, type the following on the DB2 command line:

```
db2 update dbm cfg using mon_heap_sz 1024
```

After changing this parameter, there should be enough space in the monitor heap to start taking your snapshots.

Use the following command syntax to take snapshots:

```
get snapshot for <monitor name> on <database name>
```

In the following sections, we discuss some of the snapshot monitors, describe how to interpret the output, and describe when to take action.

Buffer pool snapshot

To take a buffer pool snapshot for the central data warehouse database (usually TWH_CDW), use the **get snapshot for bufferpools on twh_cdw** command, as shown in Example 7-10.

Example 7-10 Buffer pool snapshot

```
milan:/home/db2tdw#  db2 get snapshot for bufferpools on twh_cdw

Bufferpool name                          = IBMDEFAULTBP
Database name                            = TWH_CDW
Database path                            =
/cdw_data/db2tdw/NODE0000/SQL00001/
Input database alias                     = TWH_CDW
Snapshot timestamp                       = 09-15-2004 15:51:01.407800

Buffer pool data logical reads           = 357
Buffer pool data physical reads          = 59
Buffer pool temporary data logical reads   = 0
Buffer pool temporary data physical reads  = 0
Buffer pool data writes                    = 0
Buffer pool index logical reads          = 66
Buffer pool index physical reads         = 25
Buffer pool temporary index logical reads  = 0
Buffer pool temporary index physical reads = 0

<SNIP>
```

```
Vectored IOs                                = 2
Pages from vectored IOs                     = 59
Block IOs                                   = 0
Pages from block IOs                        = 0
Physical page maps                          = 0

Node number                                 = 0
Tablespaces using bufferpool                = 3
Alter bufferpool information:
 Pages left to remove                       = 0
 Current size                               = 1000
 Post-alter size                            = 1000
```

Note: The output shown in Example 7-9, Example 7-10, and Example 7-11 is not complete. Only relevant sections are listed. Use the snapshot command displayed in the example to see the full output.

To determine the efficiency of a buffer pool, you want to calculate its buffer pool hit ratio (BPHR). The information you need to calculate the BPHR is in bold in Example 7-7 on page 197. Ideally, the BPHR would be 90% or higher. Use the following formula to calculate the BPHR:

BPHR (%) = (1- ((“Buffer pool data physical reads” + “Buffer pool index physical reads”) / (“Buffer pool data logical reads” + “Buffer pool index logical reads”))) * 100

Using the numbers from Example 7-9 on page 198, we can calculate the BPHR as follows:

= (1- ((59 + 25) / (357+66))) * 100
= (1- (84 / 423)) * 100
= (1- 0.1986) * 100
= 80.14%

In this case, the BPHR is about 80%. Currently, the IBMDEFAULTBP buffer pool is only 1000 * 4 KB pages (4 MB). By increasing this buffer pool and running the query again, we can see if the BPHR increases. If the BPHR remains low, it is advisable to reevaluate your logical design, as discussed in 7.1.2, “Buffer pools” on page 176 and 7.1.3, “Tablespaces” on page 179.

If this is a block-based buffer pool, and you are seeing a low number of “block I/Os,” consider altering the buffer pool and increasing the size of the NUMBLOCKPAGES. If you see more “block I/Os” occurring as a result, consider increasing again. If you see diminishing returns, decrease the size.

Database manager snapshot

To take a database manager snapshot, use the **`get snapshot for database manager`** command, as shown in Example 7-11.

Example 7-11 Database manager snapshot

```
milan:/home/db2tdw#  db2 get snapshot for database manager

            Database Manager Snapshot

Node type                                      = Enterprise Server Edition with
local and remote clients
Instance name                                   = db2tdw
Number of database partitions in DB2 instance  = 1
Database manager status                         = Active

Product name                                    = DB2 v8.1.1.48
Service level                                   = s040212 (U496793)

Private Sort heap allocated                     = 0                (SHEAPTHRES
Private Sort heap high water mark               = 48000
Post threshold sorts                            = 0                      and
Piped sorts requested                           = 50
Piped sorts accepted                            = 50                  SORTHEAP)

Start Database Manager timestamp                = 09-15-2004 15:34:13.066979
Last reset timestamp                            =
Snapshot timestamp                              = 09-15-2004 16:59:35.922993

Remote connections to db manager                = 0
Remote connections executing in db manager      = 0
Local connections                               = 1
Local connections executing in db manager       = 0
Active local databases                          = 1                     (NUMDB)
```

We explain and analyze the areas in bold in the following sections.

SHEAPTHRES

The important section in this snapshot is the “sort” section. These parameters are all related to the SHEAPTHRES database manager configuration parameter and the SORTHEAP database configuration parameter. The default value of 20000 will most likely be too small for your environment. Set SHEAPTHRES to a value between 40000 and 60000.

When “Piped sorts accepted” is a low value compared to “Piped sorts requested,” performance can typically be improved by increasing the size of SHEAPTHRES. If “Post threshold sorts” is a high value (that is, double-digit numbers), try

Archived

increasing the size of SHEAPTHRES. “Total Private Sort heap allocated” (found in the database snapshot output) should be less than SHEAPTHRES. If it is not, increase SHEAPTHRES.

NUMDB

Another parameter that can be tuned is NUMDB. This parameter specifies how many databases can be active in the instance. The default is 8. Set this to the amount of active databases in your instance, for example, 2, TWH_CDW and TWH_MART.

Database snapshot

Taking a database snapshot is very similar to the database manager snapshot. Use the **get snapshot for database on <database name>** command, as shown in Example 7-12 to take the snapshot.

Example 7-12 Database snapshot

```
istanbul:/home/db2tdw#  db2 get snapshot for database on twh_cdw

              Database Snapshot

Database name                          = TWH_CDW
Database path                          =
/home/db2tdw/db2tdw/NODE0000/SQL00003/
Input database alias                   = TWH_CDW
Database status                        = Active
Catalog database partition number      = 0
Catalog network node name              =
Operating system running at database server= AIX
Location of the database               = Local
First database connect timestamp       = 09/15/2004 17:04:40.984042
Last reset timestamp                   =
Last backup timestamp                  =
Snapshot timestamp                     = 09/15/2004 17:15:54.505959

High water mark for connections        = 2
Application connects                   = 6
Secondary connects total               = 0
Applications connected currently       = 1                    (AVG_APPLS)
Appls. executing in db manager currently   = 0
Agents associated with applications    = 1
Maximum agents associated with applications= 2
Maximum coordinating agents            = 2

Locks held currently                   = 0
Lock waits                             = 0
Time database waited on locks (ms)     = 0
```

```
Lock list memory in use (Bytes)            = 960       (LOCKLIST and MAXLOCKS)
Deadlocks detected                         = 0
Lock escalations                           = 0         (LOCKLIST and MAXLOCKS)
Exclusive lock escalations                 = 0         (LOCKLIST and MAXLOCKS)
Agents currently waiting on locks          = 0
Lock Timeouts                              = 0
Number of indoubt transactions             = 0

Total Private Sort heap allocated          = 0       (SHEAPTHRES and SORTHEAP)
Total Shared Sort heap allocated           = 0   (SHEAPTHRES_SHR and SORTHEAP)
Shared Sort heap high water mark           = 0   (SHEAPTHRES_SHR and SORTHEAP)
Total sorts                                = 0
Total sort time (ms)                       = 0
Sort overflows                             = 0                     (SORTHEAP)
Active sorts                               = 0

Buffer pool data logical reads             = 0
Buffer pool data physical reads            = 0
Buffer pool temporary data logical reads   = 0
Buffer pool temporary data physical reads  = 0
Asynchronous pool data page reads          = 0                (NUM_IOSERVERS)
Buffer pool data writes                    = 0 (CHNGPGS_THRESH NUM_IOCLEANERS)
Asynchronous pool data page writes         = 0 (CHNGPGS_THRESH NUM_IOCLEANERS)
Buffer pool index logical reads            = 0
Buffer pool index physical reads           = 0
Buffer pool temporary index logical reads  = 0
Buffer pool temporary index physical reads = 0
Asynchronous pool index page reads         = 0                (NUM_IOSERVERS)
Buffer pool index writes                   = 0 (CHNGPGS_THRESH NUM_IOCLEANERS)
Asynchronous pool index page writes        = 0 (CHNGPGS_THRESH NUM_IOCLEANERS)
Total buffer pool read time (ms)           = 0
Total buffer pool write time (ms)          = 0
Total elapsed asynchronous read time       = 0
Total elapsed asynchronous write time      = 0
Asynchronous data read requests            = 0
Asynchronous index read requests           = 0
No victim buffers available                = 0
LSN Gap cleaner triggers                   = 0
Dirty page steal cleaner triggers          = 0               (CHNGPGS_THRESH)
Dirty page threshold cleaner triggers      = 0               (CHNGPGS_THRESH)
Time waited for prefetch (ms)              = 0                (NUM_IOSERVERS)
Unread prefetch pages                      = 0
Direct reads                               = 0
Direct writes                              = 0
Direct read requests                       = 0
Direct write requests                      = 0
Direct reads elapsed time (ms)             = 0
Direct write elapsed time (ms)             = 0
Database files closed                      = 0                     (MAXFILOP)
```

Archived

```
Data pages copied to extended storage      = 0
Index pages copied to extended storage     = 0
Data pages copied from extended storage    = 0
Index pages copied from extended storage   = 0

Host execution elapsed time                = Not Collected

Rows deleted                               = 0
Rows inserted                              = 0
Rows updated                               = 0
Rows selected                              = 1
Rows read                                  = 20          (table scans/indexes!)
Binds/precompiles attempted                = 0

Package cache lookups                      = 11                    (PKGCACHESZ)
Package cache inserts                      = 1                     (PKGCACHESZ)
Package cache overflows                    = 0                     (PKGCACHESZ)
Package cache high water mark (Bytes)      = 138560                (PKGCACHESZ)
Application section lookups                = 12
Application section inserts                = 1

Catalog cache lookups                      = 35                (CATALOGCACHE_SZ)
Catalog cache inserts                      = 8                 (CATALOGCACHE_SZ)
Catalog cache overflows                    = 0                 (CATALOGCACHE_SZ)
Catalog cache high water mark              = 0                 (CATALOGCACHE_SZ)

Number of hash joins                       = 0
Number of hash loops                       = 0
Number of hash join overflows              = 0                       (SORTHEAP)
Number of small hash join overflows        = 0                       (SORTHEAP)
```

We explain and analyze the areas in bold in the following sections.

ROWS READ

We start with a snapshot line that is not influenced by any database or database manager configuration parameter but is very important for getting the most out of your Tivoli Data Warehouse V1.3 environment.

If the amount of rows read is very big (millions) the cause could be that the database manager is performing full table scans to get to the result of a query. Perform regular table snapshots to determine which tables are referenced the most and dynamic SQL statement snapshots to find out which query is the “villain” as follows:

```
get snapshot for tables on <database name>
get snapshot for dynamic sql on <database name>
```

After you found the problem, you can then use the Design Advisor to find out whether an index would solve the problem or not.

AVG_APPLS

This parameter defines the average number of active applications. It helps the optimizer decide how many buffer pool pages will be available at runtime for an access plan. Keep this number low. The default is 1.

LOCKLIST and MAXLOCKS

Locklist is an area in memory where DB2 stores its locks held by all applications currently connected to the database. Maxlocks determines how many locks (percentage) any connected application can store in the locklist. When maxlocks has been reached, the database manager will perform a lock escalation, where row locks for a given table will be traded in for a single table lock.

If “Lock list memory in use (Bytes)” exceeds 50% of the defined locklist size, increase the locklist. If there are “Lock escalations” or “Exclusive lock escalations” occurring, increase either locklist or maxlocks, or both.

SORTHEAP

This parameter is extremely important in your Tivoli Data Warehouse V1.3 environment. It specifies the maximum number of private memory pages to be used for private sorts or the maximum number of shared memory pages to be used for shared sorts.

A good setting to start with is 8192. If there are a lot of “Sort overflows” (double-digit numbers), increase your sortheap. If the “Number of hash join overflows” is not 0, or if the “Number of small hash join overflows” is not 0, also increase your sortheap.

CHNGPGS_THRESH

This parameter specifies the percentage of allowed changed pages in the buffer pool before the asynchronous page cleaners will be started to write the changes to disk. This is done to make room for new data in the buffer pool.

If “Dirty page steal cleaner triggers” is a double-digit number, try lowering this parameter. If “Buffer pool data writes” and “Asynchronous pool data page writes” are high, try lowering this parameter.

As mentioned in 7.1.5, “Environment settings and the Configuration Advisor” on page 191, chngpgs_thresh will be ignored if you are using DB2_USE_ALTERNATE_PAGE_CLEANING=YES.

Archived

NUM_IOCLEANERS

This specifies the number of asynchronous page cleaners for a database. These page cleaners write changed (dirty) pages from the buffer pool to disk. Start by setting it equal to the number of CPUs on the system.

Decrease num_iocleaners if Asynchronous Write Percentage (AWP) is 90% or higher, and increase if less than 90%. Calculate the AWP with the following formula:

> ((“Asynchronous pool data page writes” + “Asynchronous pool index page writes”) * 100) / (“Buffer pool data writes” + “Buffer pool index writes”)

NUM_IOSERVERS

This parameter is also referred to as prefetchers. This parameter specifies the maximum number of I/O servers for a database. Try setting it initially to the number of physical disks on which the database resides plus 1 or 2, but not more than 4-6 times the number of CPUs.

If you see “Time waited for prefetch (ms)” in the seconds, try adding an I/O server to see if performance improves.

MAXFILOP

The maxfilop parameter determines the maximum number of files that DB2 can have open concurrently. After it reaches that point, DB2 will start closing and opening its files. Opening and closing files slows SQL response and burns CPU cycles. To find out if DB2 is closing files, look for “Database files closed.” If the number is greater than 0, increase MAXFILOP.

LOGBUFSZ

This parameter is used to buffer log records before writing them to disk. The log records are written to disk when a transaction commits or the log buffer is full. Start with 256 and increase when you consistently see more than a couple of “Log pages read.”

PKGCACHESZ

Package cache is used for caching sections of SQL statements. Caching packages allows the database manager to reduce disk I/O by eliminating the need to access the system catalogs when reloading or recompiling packages.

Increase it if the value of “Package cache overflows” is greater than 0. The package cache hit ratio (PCHR) should be as close to 100% as possible. Calculate this with the following formula:

> PCHR= (1- (“Package cache inserts” / “Package cache lookups”)) * 100

CATALOGCACHE_SZ

When preparing execution strategies for SQL statements, DB2 checks this cache to learn about the definition of the database, tablespace, tables, indexes, and views. If all the information is available in memory, DB2 can avoid disk I/O, which shortens prepare time.

Increase this, if the value of “Catalog cache overflows” is greater than 0. The catalog cache hit ratio (CCHR) should be around 90-95%. Use the following formula:

CCHR= (1- (“Catalog cache inserts” / “Catalog cache lookups”)) * 100

Other parameters

There are other parameters that very important to the performance of your DB2 system. We list these in the following sections, specifying whether they are database manager configuration parameters or database configuration parameters.

INTRA_PARALLEL (database manager)

When enabled, this parameter will allow DB2 to use multiple CPUs to do its processing. Because of the complex SQL used in the ETLs, your Tivoli Data Warehouse V1.3 will most likely benefit from this. Carefully test and record runtimes to see whether or not your ETLs are running faster with intra_parallel enabled.

DFT_DEGREE (database)

The intra_parallel parameter does not work without changing the dtf_degree parameter. Set this parameter to -1 to let DB2 compute the degree of parallelism based on the number of system CPUs and other available resources.

Important: When you enable intra_parallel, make sure to set max_querydegree (database manager) equal to the number of CPUs on the system. Doing so prevents users from “accidently” inflating the requested degree of parallelism.

DFT_QUERYOPT (database)

This parameter is used to specify the level of optimization to use when compiling SQL queries. The default is class 5. Your environment might benefit from another level. Try using classes 3 or 7, but avoid using class 9.

Archived

7.1.7 Maintenance

So far, we have talked about buffer pools, tablespaces, indexes, and configuration parameters needed to tune your Tivoli Data Warehouse V1.3 environment. We also discussed the need for monitoring your databases in order to find tuning “hot spots.”

The last, but very important, task is database maintenance. If you do not run maintenance tasks from time to time, you will see that the performance of your database degrades. Before the introduction of DB2 V8.2, these maintenance tasks had to be run manually (or scheduled, using, for example, the DB2 task center or UNIX cron). This always meant human interaction was needed.

DB2 V8.2 introduces the capability to fully automate these maintenance tasks. We discuss two of these tasks that are important for your Tivoli Data Warehouse V1.3 environment.

- Automatic statistics collection attempts to improve the performance of the database by maintaining up-to-date table statistics. The goal is to allow the optimizer to choose an access plan based on accurate statistics.
- Automatic reorganization manages offline table and index reorganization without users having to worry about when and how to reorganize their data.

We discuss these two tasks in detail in the following sections.

Automatic statistics collection

When the SQL compiler optimizes SQL query plans, its decisions are heavily influenced by statistical information about the size of the database tables and indexes. The optimizer also uses information about the distribution of data in specific columns of tables and indexes if these columns are used to select rows or join tables. The optimizer uses this information to estimate the costs of alternative access plans for each query. When significant numbers of table rows are added or removed, or if data in columns for which you collect statistics is updated, the RUNSTATS utility needs to be run again to update the statistics.

Automatic statistics collection works by determining the minimum set of statistics that give optimal performance improvement. The decision to collect or update statistics is taken by observing and learning how often tables are modified and how much the table statistics have changed. The automatic statistics collection algorithm learns over time how fast the statistics change on a per table basis and internally schedules RUNSTATS execution accordingly.

Normal database maintenance activities, such as when a user performs RUNSTATS, REORG, or altering or dropping the table, are not affected by the enablement of this feature.

Automatic reorganization

After many changes to table data, logically sequential data might be on non-sequential physical pages, so the database manager has to perform additional read operations to access data.

Among other information, the statistical information collected by RUNSTATS shows the data distribution within a table. In particular, analysis of these statistics can indicate when and what kind of reorganization is necessary. Automatic reorganization determines the need for reorganization on tables by using the REORGCHK formulas. It periodically evaluates tables that have had their statistics updated to see if reorganization is required. If so, it internally schedules a classic reorganization for the table. This requires that your applications function without write access to the tables being reorganized.

Note: If you do not want to automate these tasks, make sure you manually perform the RUNSTATS and REORG commands periodically.

Enabling automatic maintenance

To enable these automated tasks, perform the following steps:

1. Select **Start** → **IBM DB2** → **General Administration Tools** → **Control Center**.
2. In the DB2 Control Center window, select **All Databases** to display the database list.
3. Right-click the **TWH_CDW** database icon, and select **Configure Automatic Maintenance**.
4. The Configure Automatic Maintenance window opens. Click **Next** on the Introduction page.
5. On the Type page, select **Change automation settings**, and click **Next** to go to the Timing page.

   The Timing page is an important page, because this is where you define your maintenance windows. Here, we see two different types of maintenance:

   – Online maintenance

   – Offline maintenance

   Online maintenance is maintenance that can be done while users are connected to the database. Offline maintenance is the opposite, which means that users are not allowed to be connected to the database when these maintenance tasks run.

   The default value for online automatic maintenance (runstats) is 00:00 - 23:00 hours, ALL days of the week, ALL days of the month. You can change this to

fit your environment. There is no default value for offline maintenance. To define the offline maintenance window, click the **Change** button and define your offline maintenance window settings.

Figure 7-5 shows an offline maintenance window from 3 a.m. to 7 a.m.

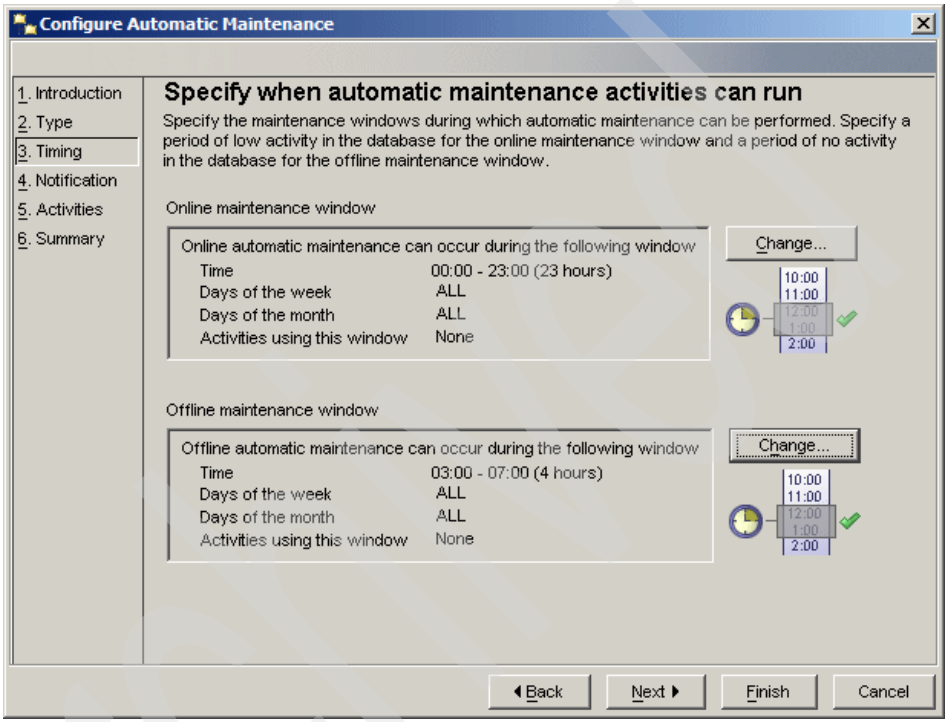


Figure 7-5 Offline maintenance window

6. Click **Next** to go to the Notification page. Select the person/group you would like to have notified, and click **Next** to go to the Activities page.
7. On the Activities page, select the activities to automate. Figure 7-6 on page 211 shows that we selected to automate REORG (offline) and RUNSTATS (online).

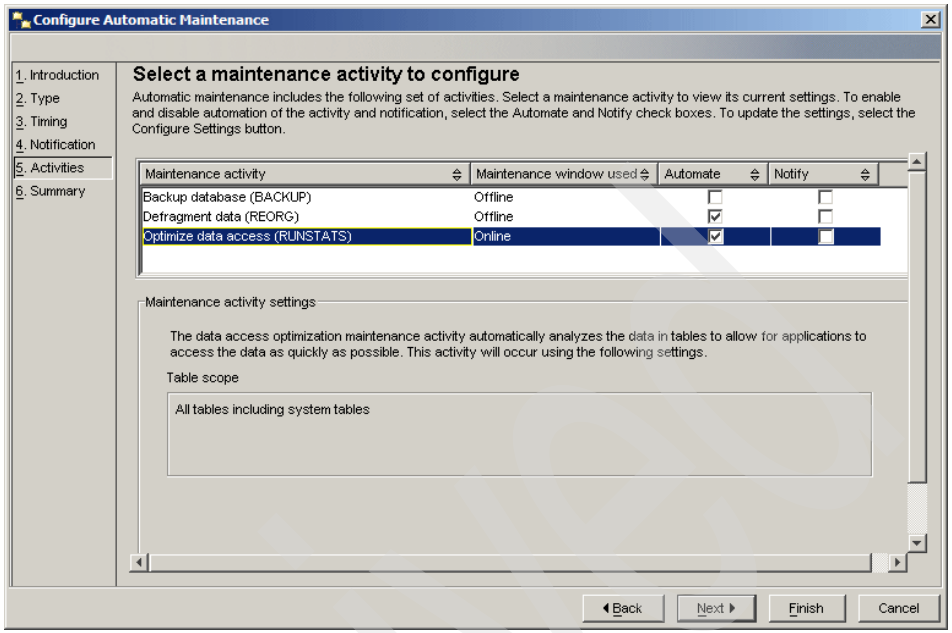


Figure 7-6 Selecting activities

8. Click **Next** to go the Summary page. Click **Finish** to enable automatic maintenance.

Important: These automated maintenance activities do *not* occur at scheduled times. Maintenance will *only* occur when DB2 determines that it is required *and* a maintenance window is available.

Manual statistics collection and reorganization

For a DB2 administrators who wants to control statistics collections and reorganization themselves, you can run REORG and RUNSTATS manually.

To identify if tables need to be reorganized, type:

```
reorgchk update statistics on table all
reorgchk current statistics on table all
```

Use `update statistics` to update the statistics before running reorg. Use `current statistics` to use the current statistics.

Tables that need to be reorganized will show one or more asterisks (*) in the reorg columns of the REORGCHK output. After you have identified which tables need to be reorganized, type:

```
reorg table <tableschema>.<tablename>
```

To order a table in a physical sequence according to a specific index, type:

```
reorg table <tableschema>.<tablename> index <indexschema>.<indexname>
```

To perform a reorganization on all the indexes for a table, type:

```
reorg indexes all for table <tableschema>.<tablename>
```

After reorganizing tables and indexes, you should always recollect the table and index statistics so that the optimizer can create the best access plan.

A typical RUNSTATS command would be:

```
runstats on table <tableschama>.<tablename> with distribution and detailed
indexes all
```

Note: Both utilities in the previous examples will run in offline mode. To run these utilities in online mode, add **`allow read access`** to the commands for read-only access. Runstats also allows read and write access by replacing **`allow read access`** with **`allow write access`**.

After performing REORG and RUNSTATS, it is advised to REBIND all database packages so that their static SQL can take advantage of the most recent system statistics.

To rebind all packages, type:

```
db2rbind -l logfile.out ALL
```

7.2 Operating system performance tuning

Besides monitoring and tuning your database environment, it is also important to do the same for your operating system. In this section, we discuss some of the monitoring tools on Windows and AIX that will help you identify possible performance bottlenecks.

We do not, however, discuss how to actually tune your operating system. This is a complex task, and separate Redbooks exist about this topic.

For AIX performance tuning, refer to *IBM @server Certification Study Guide - AIX 5L Performance and System Tuning*, SG24-6184.

Archived

For Windows performance tuning, there many books in the market that cover this topic. In addition, you can use Internet search engines to find information about Windows tuning.

7.2.1 Windows environments

An easy way of monitoring system performance on a Windows system is using the Performance Monitor and the Task Manager. The Performance Monitor enables you to obtain detailed information about many different objects, such as memory, CPU, disk, and network utilization. The Task Manager provides a quick overview of what is occurring on your system. We discuss the Performance Monitor only, because most users are already familiar with the Task Monitor.

To start the Performance Monitor, complete the following steps:

1. Select **Start** → **Run**.
2. Type `perfmon`.
3. Click **OK**.

Before we can monitor any objects, we need to add them manually. To do this, click the "**+**" button in the middle of the Performance Monitor window or press Ctrl+I. The Add Counters window opens, as shown in Figure 7-7.

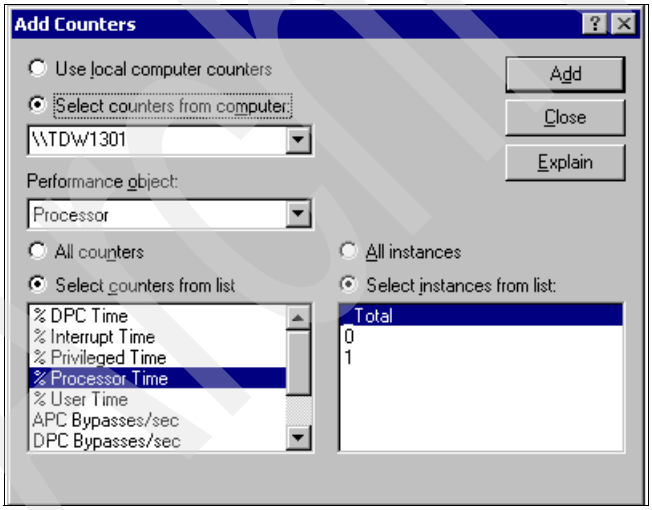


Figure 7-7 Add perfmon counters

To add counters to the Performance Monitor, complete the following steps:

1. Select the Performance object from the drop-down list.
2. Select a counter from the list.

3. If there are any instances (for example, CPU0 and CPU1, or DISK0, DISK1, and DISK2), either select **_Total** for an average, or select the individual instances for a detailed view.
4. Click **Add** to add the counter.
5. Click **Close** to go back to the Performance Monitor window.

Most of these counters are not that interesting for your Tivoli Data Warehouse V1.3 environment, but there are some that can help you spot bottlenecks on your system. These are listed in Table 7-5.

Table 7-5 Perfmon objects

Performance object	Counter
Processor	% Processor time
Physical Disk	% Disk time
Paging File	% Usages

Adding these counters to your Performance Monitor will enable you to monitor disk usage, processor usage, and page file usage.

Tip: During installation of DB2 on a Windows system, DB2 performance objects will automatically be added to the Performance Monitor. These performance objects and their counters are a graphical representation of DB2 snapshots. Refer to 7.1.6, “Snapshot monitoring” on page 197 for a detailed description of DB2 snapshots.

Figure 7-8 on page 215 shows the main Performance Monitor window.

Archived

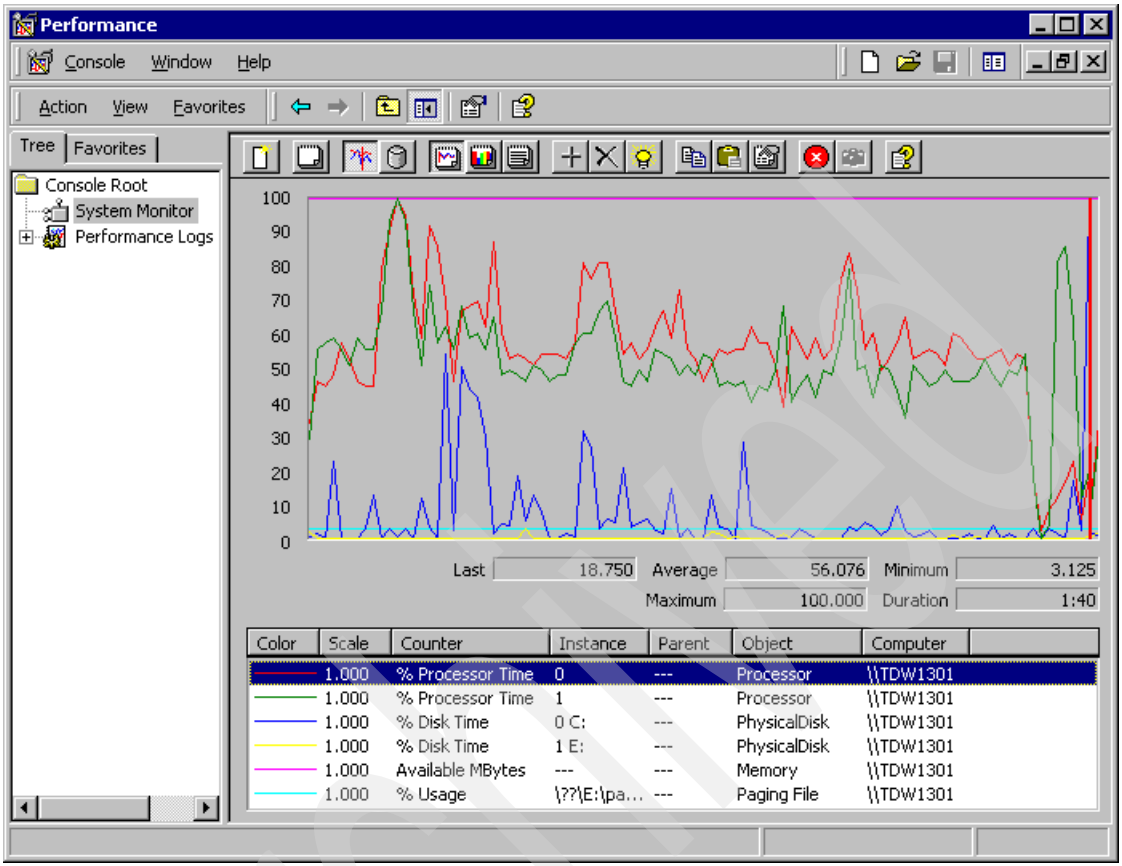


Figure 7-8 Windows Performance Monitor

Each of the selected counters are shown at the bottom of the Performance Monitor window with a different color code per counter. To highlight a specific counter, select the counter and click the light bulb icon in the middle of the window or press Ctrl+H.

7.2.2 AIX environments

The AIX operating system has several tools to monitor its resources. We discuss some of these tools in the following sections.

CPU and memory utilization: vmstat

The **vmstat** command reports statistics about kernel threads, virtual memory, disks, traps, and CPU activity. Reports generated by the **vmstat** command can be used to balance system load activity. These system-wide statistics (among all

processors) are calculated as averages for values expressed as percentages, and as sums otherwise.

Example 7-13 shows an example of the **vmstat** command with the interval and count parameters.

Example 7-13 Example vmstat command

```
# vmstat 1 5
kthr     memory             page              faults        cpu
----- ----------- ------------------------ ------------ -----------
 r  b   avm   fre  re  pi  po  fr   sr  cy  in   sy  cs us sy id wa
 0  0 15982  1388   0   0   0   8   22   0 113  281  36  1  0 98  1
 0  0 15982  1387   0   0   0   0    0   0 108 4194  31  2  3 95  0
 0  0 15982  1387   0   0   0   0    0   0 109  286  30  0  0 99  0
 0  0 15982  1387   0   0   0   0    0   0 108  285  26  0  0 99  0
 0  0 15982  1387   0   0   0   0    0   0 111  286  32  0  0 99  0
```

The **vmstat** output is shown five times with a one second interval.

The kthr column provides information about the average number of threads on various queues. The memory column shows information about real and virtual memory. Information about page faults and paging activity is shown in the page column. The faults column provides information about process control. The last column, cpu, provides a breakdown of CPU usage.

CPU and disk utilization: iostat

The **iostat** tool is typically used to find I/O bottlenecks on your system. We use Figure 7-9 to explain the **iostat** output.

```
y:        tin          tout    avg-cpu:  % user     % sys      % idle     % iowai
          2.5         463.0                3.4       1.5        44.9       50.2

sks:          % tm_act      Kbps       tps    Kb_read   Kb_wrtn
isk0           68.0       508.0      83.0          0      1016
isk1            1.0         6.0       1.5          4         8
isk2           56.0       352.0      66.0          0       704
isk6            0.0         0.0       0.0          0         0
isk7            0.0         0.0       0.0          0         0
isk8            0.0         0.0       0.0          0         0
isk3           19.0       176.0      25.5          0       352
isk4           20.5       192.0      26.0          0       384
isk9            0.0         0.0       0.0          0         0
isk5           49.0       352.0      61.5          0       704
0               0.0         0.0       0.0          0         0
```

Figure 7-9 iostat sample output

The CPU statistics columns (% user, % sys, % idle, and % iowait) provide a breakdown of CPU usage. This information is the same as the CPU values reported in the output of the **vmstat** command in the columns labeled us, sy, id, and wa.

The % iowait column is very important for your Tivoli Data Warehouse V1.3 environment. This column shows the percentage of time the CPU was idle with pending local disk I/O. This can have several causes, for example:

- Inefficient I/O subsystem configuration
- High workload

Both the inefficient I/O subsystem configuration and the high workload are very much related to the physical layout of your database or databases. In 7.1.3, “Tablespaces” on page 179, we talk about this subject and how important it is to spread data across as many physical disks as possible.

The high percentage of I/O wait (50%), as shown in Figure 7-9 on page 216, could mean that there are not enough disks available to handle the workload. To be able to find proof of this, we need to find out how busy the disks are.

High disk activity can be identified by the % tm_act column. It shows the percentage of time the volume was active. It is the primary indicator of a bottleneck.

The high activity percentage on hdisk0 (68%) in Figure 7-9 on page 216 tells us that it would be a good idea to add another disk to the logical volume. The screen capture was taken when running an ETL on one of our test lab machines. The hdisk0 device was used for DB2 transaction logging, proving the importance of separating the DB2 transaction logs from the data and indexes.

When you see a high iowait percentage and a high disk activity on one (or more) of your disks, consider adding a disk to reduce I/O contention.

The syntax for the iostat command is:

```
iostat [interval] [count]
```

For example:

```
iostat 2 10
```

This will show you the output 10 times with a 2 second interval.

Overall utilization: topas

This real-time tool enables you to see many different resource utilization figures. It is part of the AIX Performance Tools and has to be installed separately. Two

packages are needed to install it: perfagent.tools and bos.perf.tools. It combines output from the **sar**, **vmstat**, **iostat**, and **netstat** commands into one screen.

Figure 7-10 shows a typical topas screen.

```
Topas Monitor for host:    milan                  EVENTS/QUEUES    FILE/TTY
Thu Sep  2 10:43:23 2004   Interval:  2           Cswitch    4164  Readch    12288
                                                  Syscall    3385  Writech 2684.6K
CPU       User%   Kern%   Wait%   Idle%           Reads         3  Rawin         0
cpu1       21.0     5.5    11.0    62.5           Writes      671  Ttyout      656
cpu2       21.0     6.5    34.0    38.5           Forks         0  Igets         0
cpu3       18.5     4.5    16.5    60.5           Execs         0  Namei         2
cpu0       15.0     5.5    68.5    11.0           Runqueue    0.0  Dirblk        0
                                                  Waitqueue   2.0
Network   KBPS   I-Pack  O-Pack   KB-In  KB-Out
en0        1.7     13.0     8.0     1.4     2.1  PAGING            MEMORY
lo0        0.0      0.0     0.0     0.0     0.0  Faults       34   Real,MB    1023
                                                  Steals      702   % Comp     69.9
Disk     Busy%     KBPS     TPS KB-Read KB-Writ  PgspIn        0   % Noncomp  31.0
hdisk2    98.0   2644.0   170.0     4.0  5284.0  PgspOut       0   % Client    0.5
hdisk3    93.0   2700.0   164.0     0.0  5400.0  PageIn        0
hdisk5    92.0   2726.0   162.5     0.0  5452.0  PageOut    2676   PAGING SPACE
hdisk4    82.5   2678.0   154.5     0.0  5356.0  Sios       1097   Size,MB     512
hdisk0     0.5      4.0     1.0     0.0     8.0                     % Used     32.8
hdisk8     0.0      0.0     0.0     0.0     0.0  NFS (calls/sec)    % Free     67.1
cd0        0.0      0.0     0.0     0.0     0.0  ServerV2       0
hdisk6     0.0      0.0     0.0     0.0     0.0  ClientV2       0    Press:
hdisk7     0.0      0.0     0.0     0.0     0.0  ServerV3       0    "h" for help
hdisk9     0.0      0.0     0.0     0.0     0.0  ClientV3       0    "q" to quit
hdisk1     0.0      0.0     0.0     0.0     0.0

Name            PID  CPU%  PgSp Owner
db2sysc       53158  12.2   5.7 db2tdw
db2sysc       54446   2.4   5.4 db2tdw
db2sysc       46712   2.0   5.5 db2tdw
db2sysc       67594   1.6   6.5 db2tdw
db2sysc       47308   0.9   5.5 db2tdw
```

Figure 7-10 Sample topas output

There are several things we can learn from the output shown in Figure 7-10:

- Relatively high iowait percentage (wait%)
- Very high disk usage (busy%)

As with Figure 7-9 on page 216, Figure 7-10 shows that we have a major disk problem. These disks (hdisk2, 3, 4, and 5) are 90%+ busy, showing that adding more disks to improve performance is no luxury.

To start **topas**, simply type **topas** on the command line. When you have an SMP machine, press c twice to show all CPUs. Type h to see other options.

7.3 Tivoli Data Warehouse performance

The overall performance of the Tivoli Data Warehouse will be very much dependent on how well the hardware, operating system, and DB2 environment has been configured. Next, we discuss a number of Tivoli Data Warehouse components that can be tuned.

Distributed install

The Tivoli Data Warehouse components can be, but do not need to be, installed on the same systems as other Tivoli software or on the systems where the operational data stores reside.

The operational data stores are on a system that is not part of the Tivoli Data Warehouse deployment. As an example, the Tivoli Enterprise Console database will have operational data.

Crystal Enterprise should reside on a system that does not have other Tivoli Data Warehouse or Tivoli software products on it. Users access Crystal Enterprise reports using a Web browser from any system.

Other types of data analysis tools should be located on systems outside your Tivoli Data Warehouse deployment.

Control server and warehouse agent

The DB2 warehouse agent is the component of IBM DB2 Warehouse Manager that manages the flow of data between the data sources and targets that are on different computers. By default, the control server uses a local warehouse agent to manage the data flow between the operational data sources, central data warehouse databases, and data mart databases.

In some environments, this configuration is sufficient. Optionally, you can create agent sites on other Windows and UNIX systems in your environment. The control server can use the agents on these computers to manage the data flow. In a distributed deployment, you can improve the performance of Tivoli Data Warehouse by creating an agent site on the computer that is the target of the data transfer for a central data warehouse or data mart ETL routine.

That computer becomes a remote agent site, which the Data Warehouse Center uses to manage the transfer of Tivoli Data Warehouse data. This can speed up the data transfer, as well as reduce the workload on the control server.

The control server is the system that contains the control database for Tivoli Data Warehouse and is the system from which you manage your data. The control database contains metadata for both Tivoli Data Warehouse and for the warehouse management functions of IBM DB2 Universal Database Enterprise

Edition. You can have only one control server in a Tivoli Data Warehouse deployment.

On Windows and UNIX systems, using a warehouse agent on the computer that contains the central data warehouse database or a data mart database can potentially improve performance.

ETL routines

The scheduling of data warehouse ETLs should be done during off-peak hours to avoid impacting the performance of your operational data stores. For distributed environments across geographic locations, you might consider putting central data warehouse databases at each location, because each location might have different off-peak hours.

For example, if your operational data is on systems in the United Kingdom and the United States, you might put a central data warehouse database on a system in each location. This enables you to schedule the central data warehouse ETL for each system at an appropriate off-peak time.

The time taken for ETLs to complete depends on many factors, including the amount of data they have to process, the speed of the database in which the source and target data reside, and the performance of the network.

Ensure that the default scheduling interval is changed to an appropriate interval for your environment and data level.

The ETL processes that update tables in the central data warehouse should not all be scheduled at the same time. There might be unknown dependencies in the data, and updates to the same tables might cause performance problems, depending on your environment.

Data analysis programs can read directly from central data warehouse databases without using data marts, but this use is not supported. Analyzing historical data directly from the central data warehouse database can cause performance problems for all applications using the central data warehouse.

Archived

8

Troubleshooting

In this chapter, we discuss some of the possible problems you might encounter when working with Tivoli Data Warehouse V1.3. We cover the following topics:

- Log files
- Common installation problems
- Runtime problems
- Miscellaneous problems
- Hints and tips

© Copyright IBM Corp. 2005. All rights reserved.

8.1 Log files

Log files are an important part of your Tivoli Data Warehouse V1.3 environment. They can help you find the cause of a problem and might even help you solve that problem.

There are several types of logs:

- Tivoli Data Warehouse V1.3 installation logs
- Warehouse pack installation logs
- Runtime (ETL) logs

In the following sections, we outline where to find these logs and how they can potentially help you to solve your problems.

Tivoli Data Warehouse installation logs

When you first install Tivoli Data Warehouse V1.3, you will see a window regarding the Tivoli common logging directory. This will either be a notification of the common logging directory (when another Tivoli product has already been installed on the system) or a prompt to specify the common logging directory. The common logging directory will be used to log all Tivoli Data Warehouse installation activities.

By default, these log files can be found in the TIVOLI_COMMON_DIR/cdw/logs/install/* directory for both Windows and AIX:

- On Windows, the directory will most likely be:

  C:\Program Files\ibm\tivoli\common\cdw\logs\install

- On AIX, the directory is usually:

  /usr/ibm/tivoli/common/cdw/logs/install

If you experience any problems during the installation, go to these directories and locate the log files. The log files are named: core_*

These log files will all be written in an XML format. Tools for viewing Tivoli Data Warehouse logs in a more readable format can be found in the TIVOLI_COMMON_DIR/cdw/tools directory. These tools can convert the XML-formatted logs into an HTML-formatted document.

Warehouse pack installation logs

The warehouse pack installation log files are also located in the Tivoli common logging directory. The log files for warehouse pack installs are named: WP_*

Archived

Runtime logs

For every step in an ETL process, a log file will be created. These log files show every single SQL statement that was used during the runtime of the SQL. This enables the user to see what happens “under the hood” of Tivoli Data Warehouse. The file names of the log files correspond to the ETL steps in the Data Warehouse Center.

These log files can be found in the following directories:

- On Windows: %VWS_LOGGING%

  On Windows, this will typically be: C:\SQLLIB\LOGGING

- On AIX: $INSTHOME/sqllib/logging

  On AIX, this is usually: /home/db2inst1/sqllib/logging (where db2inst1 is the name of the instance)

Figure 8-1 shows the log files created during the AMX run.

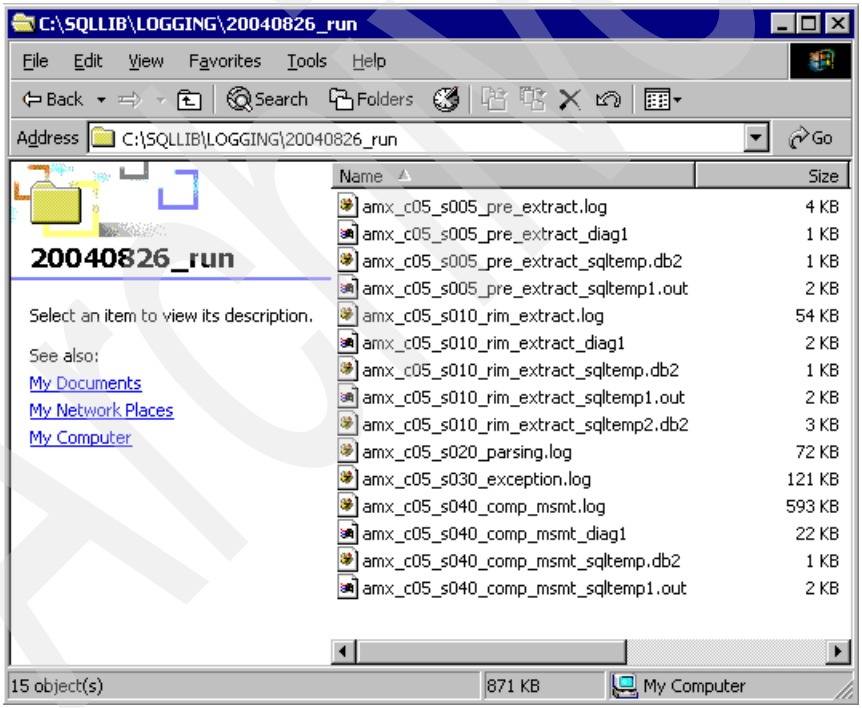


Figure 8-1 ETL log files

As you might have noticed, the directory in Figure 8-1 is not the default logging directory; it is a subdirectory of the logging directory. For testing purposes, we

copied the log files into a separate directory after each test. This allowed us to analyze the logs to monitor the impact of our changes (for example, adding an index to a table).

Many customers keep a record of these logs for similar reasons. After an ETL run finishes, the log files are copied to an archive directory where they can be analyzed at any time. In production environments, this is the perfect way to time the ETLs and detect possible tuning hot spots.

Sometimes, one of the ETL steps fails before the log file was created. If this happens to you, look for the *.diag files in the %VWS_LOGGING% directory. These files might contain information about the problem. Another option is to check the %TIVOLI_COMMON_DIR%\cdw\logs\etl\ directory.

Tip: Sort the %VWS_LOGGING% directory on the "Modified" column (Windows) to be able to see which file was created or appended last. This might help when searching for error messages. On AIX, use the `ls- ltr` command to get the same result.

8.2 Common installation problems

This section describes some of the most common installation problems. For a more detailed description of these problems, refer to the Tivoli Data Warehouse V1.3 documentation.

8.2.1 Core installation problems

The following sections describe some of the problems related to Tivoli Data Warehouse V1.3 base installations.

Installing without Crystal

When installing Tivoli Data Warehouse V1.3, the user needs to enter Crystal Enterprise server details, but does not want to use Crystal Enterprise as a reporting solution.

Problem

Installer does not want to continue without Crystal Enterprise connection details.

Solution

When the Crystal Enterprise Connection window opens during the initial installation of Tivoli Data Warehouse V1.3, type `NO_CRYSTAL` in the Host Name field. There can be no leading or trailing spaces, and the capitalization must be

exact. A value must be entered in the User Name field, but it is irrelevant and is ignored. No password must be entered.

> **Note:** Using NO_CRYSTAL during the installation will install Tivoli Data Warehouse V1.3 without native reporting capabilities. Tivoli Data Warehouse V1.3 will have to be re-installed if Crystal support is required at a later time.

Uninstall fails because of installed warehouse packs

The Tivoli Data Warehouse V1.3 installer does not allow the core to be uninstalled if any warehouse packs are installed. However, sometimes the warehouse packs are difficult to uninstall because of errors. How do we install the core without uninstalling the warehouse packs?

Solution 1

We change the state for all the warehouse packs to “inactive’ by updating a table in the metadata database. To do this, type:

```
db2 connect to twh_md user <user name> using <password>
db2 update twg.WEP_version set active_version = ‘I’, state=’D’
```

Refer to 8.2.2, “Warehouse pack installation problems” on page 225 for a more detailed explanation of the warehouse pack status.

Solution 2

Should solution 1 fail, we can try a more thorough plan. This is done by issuing the following SQL statements:

```
db2 connect to twh_md user <user name> using <password>
db2 delete from twg.WEP_role_version_profile
db2 delete from twg.WEP_install_stage
db2 delete from twg.WEP_versions
db2 delete from twg.WEP
```

These must be done in order, because the first three tables are dependent on twg.WEP.

> **Important:** Only do this when uninstalling a warehouse pack fails and thus prevents you from uninstalling core Tivoli Data Warehouse V1.3.

8.2.2 Warehouse pack installation problems

The following sections describe some of the problems related to warehouse pack installations.

Determine warehouse pack version

When applying fix packs for IBM Tivoli Monitoring and Tivoli Data Warehouse, the fix packs also need to be applied for the corresponding ETLs (AMX, AMY), and you cannot tell from the **wlsinst** command whether they are applied or not. Sometimes, we need to be sure that the latest fix pack is applied for specific ETL issues.

For Tivoli Data Warehouse V1.3, the ETL information is stored in the WP_VERSIONS table in the TWH_MD database. This table contains an entry for each release level of each ETL. One of the columns for this table is called ACTIVE_VERSION, which will contain the value “I” for inactive, or “A” for active.

Each time a new patch is applied successfully, the previous entry is marked “I”, and the new one is set to “A”. When applying a fix pack, the installer looks in this table for the latest entry marked “A”. If none are found, the installer will refuse to run.

To find the current version of your warehouse pack, issue a SQL statement on the DB2 command line. Example 8-1 shows the SQL statement.

Example 8-1 WP_version table

```
C:\>db2 select AVA_CODE, APP_REL_VERSION, APP_DISPLAY_VERSION from
twg.WP_versions where active_version = 'A'

AVA_CODE APP_REL_VERSION              APP_DISPLAY_VERSION
-------- ---------------------------- ------------------------------
AMY      1.1.0                        Version 1.1.0.600
ANM      1.1.0                        Version 1.1.0
AMX      5.1.1.670                    Version 5.1.1.670

  3 record(s) selected.
```

Warehouse logger does not start

When installing a warehouse pack, the warehouse logger service did not start. You might have received the following error:

```
CDWIC0102W The installation program failed to start the Warehouse logger
service.
```

Possible problem

The warehouse logger will not function when it cannot write to the Windows event logs.

Possible solution

Clear all events from the Windows event logs.

To do this, select **Start** → **Settings** → **Control Panel** → **Administrative Tools** → **Event Viewer**. Right-click a log, and select **Clear all Events**.

Reports cannot be installed

During the installation of a warehouse pack, you received the following error:

```
CDWIC5207E The installation program failed while trying to migrate reports
supplied with the Warehouse pack to the Crystal Enterprise format (.xml format
to .rpt format).
```

Problem

The ODBC data source on the Crystal Enterprise server has not been created. This is the connection to the data mart. It will prevent the warehouse pack from installing or migrating its reports.

Solution

Create the ODBC data source on the Crystal Enterprise server. Refer to 6.5.1, “Backing up the Tivoli Data Warehouse environment” on page 140 for a detailed description about how to create the ODBC source. After creating the ODBC source, reinstall the warehouse pack.

Failing fix pack installation

When installing a warehouse fix pack, the installation program is unable to rename a subdirectory in the warehouse pack directory, and you receive an error such as the following:

```
CDWIC5109E The installation program, could not rename the Warehouse pack's
version directory in the apps subtree to the new version number.
```

Possible problem

The directory that the installation is trying to rename might be open in Windows Explorer or are “in” the directory in a command line session. Also, a file in the directory might be open.

Solution

Move to a different directory in Windows Explorer or in your command line session, or both, and make sure that no files in that directory are open.

Archived

Recovering from a failed installation

To recover from a failed warehouse pack installation, first simply attempt to reinstall the warehouse pack that failed. If that still fails, do the following:

1. Attempt to uninstall the warehouse pack that failed.
2. Restore the TWH_MD database from a back up by typing:

```
db2stop force
dbstart
cd %TWH_TOPDIR%\apps_backups\productcode\install
db2 restore database twh_md from .
```

3. Reinstall the warehouse pack that failed.

If you are uninstalling a warehouse pack to recover from an unsuccessful installation of that same warehouse pack, you might be prompted to restore a backup copy of the control database. If you receive this prompt, you must restore the backup copy of the control database before attempting to reinstall the warehouse pack.

If you are unable to reinstall the warehouse pack, contact an IBM Support Representative.

8.3 Runtime problems

This section lists some problems that could occur when running the ETLs.

Data Warehouse Center logon fails (1)

When logging on to the Data Warehouse Center, the following error appears:

```
DWC06200E  An unexpected communications error has occurred.
RC = 6200    RC2 = 0
```

Possible problem

The warehouse server and logger Windows services are not running.

Solution 1

On the command line type:

```
net start db2dwserver
```

This will start the DB2 Data Warehouse server and logger Windows services.

Solution 2

From the Windows taskbar, select **Start** → **Settings** → **Control Panel** → **Administrative Tools** → **Services**.

Locate the **DB2 Warehouse Server service**, right-click it, and select **Start**. This will start both the DB2 Warehouse server and DB2 Warehouse logger services.

Figure 8-2 shows the Services window and the highlighted DB2 Warehouse server service.

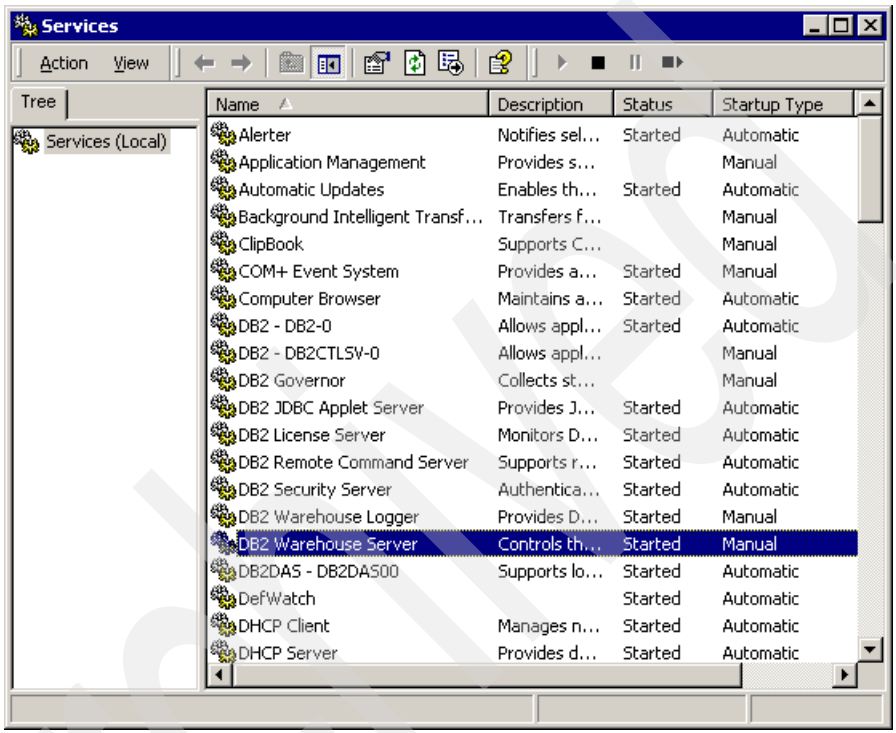


Figure 8-2 Windows services

Data Warehouse Center logon fails (2)

When logging on to the Data Warehouse Center, you receive an message similar to the following:

```
DWC07322E The Warehouse server was unable to authenticate the password provided
for user "db2admin". The error occurred in response to an authentication
request from client "mberkhou". RC = 7322    RC2 = 0
```

Possible problem

The user information in the DB2 Data Warehouse Control Database Manager is not set correctly.

Solution

Two tasks have to be performed:

1. Specify the control database (TWH_MD) for the DB2 Data Warehouse Center.

   The control database is configured during the initial installation of Tivoli Data Warehouse. Perform the procedure in this section only if you have trouble starting the DB2 Data Warehouse Center.

   To specify the control database for Tivoli Data Warehouse in the DB2 Data Warehouse Center, complete the following steps:

   a. From the Windows taskbar, select **Start** → **Programs** → **IBM DB2** → **Business Intelligence Tools** → **Data Warehouse Center**.

   b. In the Data Warehouse Center Logon window, click **Advanced**. You do not need to provide logon information. In the Advanced window, type `TWH_MD` in the Control database field, and click **OK**.

   c. In the Data Warehouse Center Logon window, type the DB2 user name and password for the control server, and then click **OK**. Select **Warehouse** → **Shut Down DB2 Tools** to close the DB2 Data Warehouse Center.

2. Specify the control database (TWH_MD) for the DB2 Warehouse Control Database Manager.

   The control database is configured during the initial installation of Tivoli Data Warehouse. Perform the procedure in this section only if you have trouble starting DB2 Data Warehouse Center.

   To specify the control database for Tivoli Data Warehouse in the DB2 Warehouse Control Database Manager, complete the following steps:

   a. Open the Data Warehouse Center - Control Database Management window by selecting **Start** → **Programs** → **IBM DB2** → **Business Intelligence Tools** → **Data Warehouse Center**.

   b. Type `TWH_MD` in the New control database field. Do not change the schema name. Type the DB2 user ID and password for the control database, and then click **OK**.

   c. When the "`Processing has completed`" message appears, click **Cancel**.

Missing ETL parameters

When running ETL1 or ETL2 in the Tivoli Data Warehouse, the following error might be encountered:

```
DWC07356E An agent's processing of a command of type "runUDP" failed for
edition "X" of step "[AVA]_Step_Name". Problem: CDW8085E: Invalid number of
parameters in [AVA]_Step_name.
```

Problem

The cause of this problem could be that one of the step parameters has not been filled in.

Solution

To solve this, open the Data Warehouse Center and expand the Data Warehouse Sources and Targets.

For each Source and Target, right-click the name, select **Properties**, **Data Source Name**, and inspect each dialog for completeness, filling in any that are missing (typically the user ID or password.)

If this does not solve the problem, the cause could be that a Tivoli Data Warehouse tag was not properly imported. In this case, it is advisable to contact an IBM Support Representative.

Cannot change ETL mode

You cannot change the ETL mode.

Problem

While working with the steps or processes from an ETL in the Data Warehouse Center GUI, it is sometimes impossible to change the mode on the ETL steps.

Solution

While this can happen legitimately because of problems with the data for the ETL in the control database (TWH_MD), before calling support, make sure that no links are selected when attempting to move the steps into a different mode.

Links are not actual ETL steps, but they are links to ETL steps in another Subject Area process. It is very easy to select all the items under a process, including the links. Figure 8-3 on page 232 shows six “regular” ETL steps and one link (AMY_c05_S010_Update). A link can be recognized by the little arrow on the icon left of the step name.

Archived

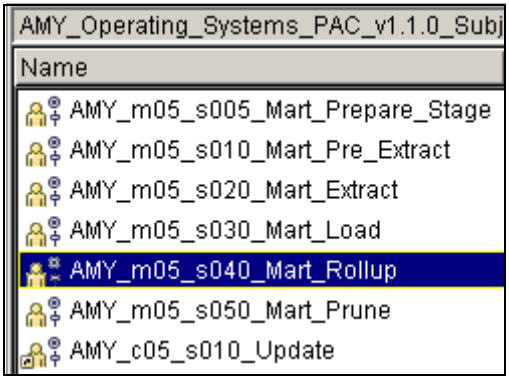


Figure 8-3 Links in the Data Warehouse Center

This will prevent you from changing the mode of the steps.

AMX Pre-Extract step fails

You try to run the AMX_c05_s005_Pre_Extract step, but it fails with the following error:

```
DWC07356E An agent's processing of a command of type "runUDP" failed for
edition "338" of step "AMX_c05_s005_Pre_Extract". AMX8267E The
AMX_c05_ETL1_Pre_Extract Step cannot be executed. All the steps of the
AMX_c05_ETL1_Process must have been executed successfully before running the
pre_extract step.
```

Possible problem

The AMX_c05_s005_Pre_Extract step does not run because you started an AMX run earlier that did not finish correctly.

Solution

Using the ETL log files, find out on which step your previous ETL run ended. Run that step again to finish the complete ETL process. Go to the Work in Progress window in the Data Warehouse Center. Select **Work in Progress** → **Run New Step**, as shown in Figure 8-4 on page 233.

Archived

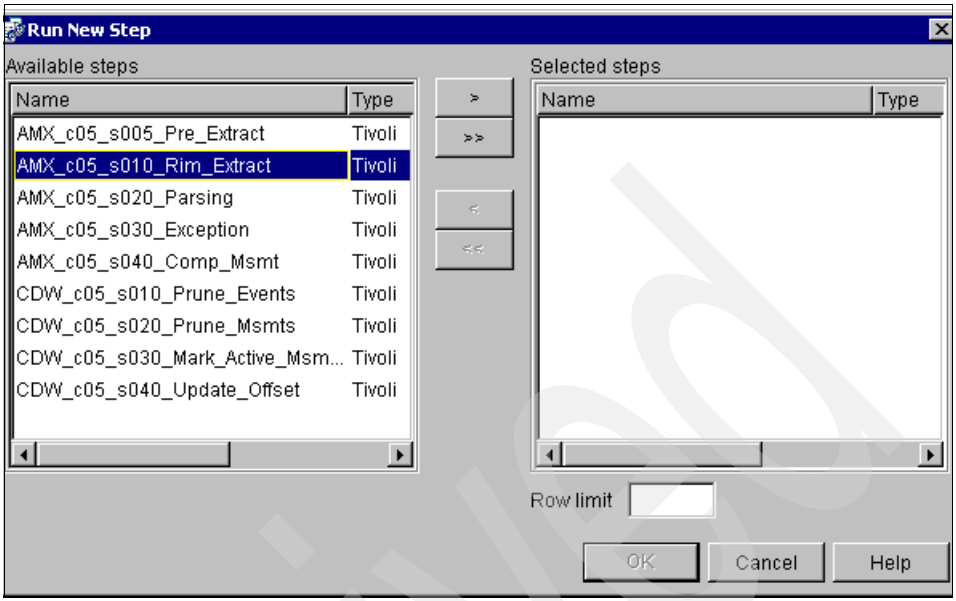


Figure 8-4 Work in Progress window

Now, highlight the step you would like to run and click the “>” button. Click **OK** to run the step.

AMX ETL runtime failure

When running the AMX ETL, it fails with the following error:

```
DWC07356E An agent's processing of a command of type "runUDP" failed for
edition "0" of step "AMX_c05_s010_Rim_Extract". AMX8267E The
AMX_c05_ETL1_Process cannot be run. At least one Warehouse Enablement Package
that requires IBM Tivoli Monitoring must be installed before running the
AMX_c05_ETL1_Process. RC = 7356    RC2 = 8410
```

Problem

The AMX warehouse pack requires the AMY warehouse pack to be installed. It will not run without the AMY warehouse pack.

Solution

Install the AMY warehouse pack.

No new data inserted during ETL run

No data is inserted into the Tivoli Data Warehouse common repository, even if new data is collected and uploaded into the IBM Tivoli Monitoring database.

Archived

Problem

The extract control functions used by the AMX_c05_ETL1 process are not aligned to the insert_seq values used to enumerate the records inserted into the IBM Tivoli Monitoring data tables. You can check this using the Extract_Win tool (in %TWH_TOPDIR%/apps/<product code>/<product version>/misc/tools) and comparing the boundaries of the last successful extraction with the current values of the insert insert_seq field.

The cause for this could be that you have rebuilt the IBM Tivoli Monitoring database from scratch or restored it from a backup performed before the last ETL1 extraction. The sequence numbers released by the RDBMS server are lower than the boundaries of the last extraction windows for the AMX warehouse pack.

Solution

Because the ETL flow depends very much on these insert sequences, we have to update some rows in the database.

Update the twg.extract_control table according to the insert_seq values of the new records in which you are interested. Do this using the AMX_Reset_Extract_Window or the Rewind_Extract scripts (both in the same tools directory previously mentioned).

8.4 Miscellaneous problems

In the following sections, we describe problems that are not specifically Tivoli Data Warehouse V1.3 problems.

Changing location of databases

After installing Tivoli Data Warehouse V1.3, you realize that the storage requirements to hold the data might be more than what the current drive can hold. Because of this, you decide that the databases need to be stored on another drive. How can the Tivoli Data Warehouse databases be created in or moved to another local drive on the same system?

Problem

Tivoli Data Warehouse V1.3 is to be installed with the databases on a different local drive or is already installed with the databases created on the default drive (for example, a quick start deployment).

Solution 1: Before installing Tivoli Data Warehouse

The location of a database is determined by the DFTDBPATH database manager configuration parameter. This current value for this parameter can be

found using the **get dbm cfg** command. It can then be updated using the **update dbm cfg** command. Example 8-2 shows you how to find the current default database path and how to change it.

Example 8-2 Database manager configuration

```
C:\tmp>db2 get dbm cfg |grep -i dftdbpath
 Default database path                        (DFTDBPATH) = C:

C:\tmp>db2 update dbm cfg using dftdbpath d:
DB20000I  The UPDATE DATABASE MANAGER CONFIGURATION command completed
successfully.

C:\tmp>db2stop
09-21-2004 16:44:56     0   0   SQL1064N  DB2STOP processing was successful.
SQL1064N  DB2STOP processing was successful.

C:\tmp>db2start
09-21-2004 16:44:59     0   0   SQL1063N  DB2START processing was successful.
SQL1063N  DB2START processing was successful.

C:\tmp>db2 get dbm cfg |grep -i dftdbpath
 Default database path                        (DFTDBPATH) = D:
```

Note: Windows only allows drive letters for the DFTDBPATH configuration parameter (for example, C:, D:). AIX will allow any path (for example, /db2data/CDW).

Solution 2: After installing Tivoli Data Warehouse

When a database has to be moved after Tivoli Data Warehouse V1.3 is already installed, things become much more complicated. A simple backup and restore to a new path is not sufficient.

During a database backup operation, a record is kept of all the tablespace containers associated with the tablespaces that are being backed up. Because the Tivoli Data Warehouse V1.3 installation program creates multiple tablespaces, a restore to a new path would only create the default tablespaces on that new path. The tablespaces that were created during installation would be restored to their original location.

To successfully move the database to another path, we have to redirect the tablespace containers from the backup image to new containers. DB2 allows this using a *redirected restore* operation.

To do this, we first need to find out which tablespaces have to be redirected. In the examples to follow, we use the TWH_CDW database.

Archived

Complete the following steps:

1. Connect to the database by typing:

   ```
   connect to TWH_CDW
   ```

2. List all the tablespaces in the database by typing:

   ```
   list tablespaces
   ```

3. Write down all the tablespace IDs except for tablespaces SYSCATSPACE, TEMPSPACE1, and USERSPACE1. These are the default tablespaces and will be moved automatically (unless you manually configured these tablespaces).

   Example 8-3 shows how to find the tablespace ID and shows where the current tablespace containers are located.

Example 8-3 List tablespaces command

```
milan:/home/db2tdw#  db2 list tablespaces

           Tablespaces for Current Database

 Tablespace ID                        = 0
 Name                                 = SYSCATSPACE
 Type                                 = System managed space
 Contents                             = Any data
 State                                = 0x0000
   Detailed explanation:
     Normal

 Tablespace ID                        = 1
 Name                                 = TEMPSPACE1
 Type                                 = System managed space
 Contents                             = System Temporary data
 State                                = 0x0000
   Detailed explanation:
     Normal

 Tablespace ID                        = 2
 Name                                 = USERSPACE1
 Type                                 = System managed space
 Contents                             = Any data
 State                                = 0x0000
   Detailed explanation:
     Normal

Tablespace ID                        = 3
 Name                                 = TWH_DATA
 Type                                 = System managed space
 Contents                             = Any data
```

```
 State                                = 0x0000

milan:/home/db2tdw#  db2 list tablespace containers for 3

            Tablespace Containers for Tablespace 3

 Container ID                         = 0
 Name                                 = /cdw_data/twh_data
 Type                                 = Path
```

In the example, we see four tablespaces with tablespace IDs 0, 1, 2, and 3. As mentioned earlier, we only have to write down tablespace ID 3. The other tablespaces are default DB2 database tablespaces. For this example and the examples to follow, we limited the output to show only one non-default tablespace (twh_data).1

4. The next step is to back up the database using the **backup database TWH_CDW to <path>** command.
5. When the backup has finished successfully, drop the database by typing:

   ```
   drop database TWH_CDW
   ```

6. To restore the database and move the tablespace containers to a new location, we use the **restore database TWH_CDW from <path> redirect** command.

   Example 8-4 shows how to do the redirected restore.

Example 8-4 Redirected restore

```
milan:/home/db2tdw#  db2 restore database TWH_CDW from /cdw_data redirect
SQL1277N  Restore has detected that one or more table space containers are
inaccessible, or has set their state to 'storage must be defined'.
DB20000I  The RESTORE DATABASE command completed successfully.

milan:/home/db2tdw#  db2 "set tablespace containers for 3 using (path
'/cdw_data/new_data')"
DB20000I  The SET TABLESPACE CONTAINERS command completed successfully.

milan:/home/db2tdw#  db2 restore database TWH_CDW continue
DB20000I  The RESTORE DATABASE command completed successfully.
```

7. To verify that our restore operation succeeded, we can issue the same commands as used in Example 8-3 on page 236. Example 8-5 on page 238 shows that the new tablespaces containers are now located on the new path as specified in the **set tablespace containers** command.

Archived

Example 8-5 List tablespaces

```
milan:/home/db2tdw#  db2 list tablespace containers for 3

            Tablespace Containers for Tablespace 3

 Container ID                         = 0
 Name                                 = /cdw_data/new_data
 Type                                 = Path
```

Tip: The redirected restore can also be used to "copy" a database to a new path on the same system by adding **INTO <new database name>** to the **restore database** command.

8.5 Hints and tips

This section describes some hints and tips that can be helpful while working with Tivoli Data Warehouse V1.3.

Command line ETLs

If, for whatever reason, you are unable to run the ETLs from the Data Warehouse Center, there is a possibility to run them manually from the command line.

The syntax for running the ETL steps manually is:

```
sqlscript.sh <product AVA code> <Full ETL step name> <source db name> <source
db user> < source db password> <target db name> <target db user> < target db
password>
```

We will use an example to explain this in more detail. In this example, we run the AMX_c05_s005_Pre_Extract step from the command line:

1. Select **Start** → **Run**. Type `bash`.
2. In the command window, type:

   ```
   sqlscript.sh AMX AMX_c05_s005_pre_extract ITM_DB itmuser password TWH_CDW
   twhuser password
   ```

Installing fix packs

Before installing a new warehouse pack, a fix pack, or a LA patch, we recommend that you back up the %TWH_TOPDIR%/apps and %TWH_TOPDIR%/apps_backups directories, and also back up the THW_CDW, TWH_MART, and TWH_MD databases.

Should anything go wrong during the installation, you can always switch back to the previous level of the installed warehouse pack.

If you have restored a backup from a previous level, make sure you remove all the dynamically generated scripts under the %TWH_TOPDIR%/apps/amx/v511/etl/sql directory so that they can be recreated using the right templates and script generator tools.

> **Note:** Note that the AMX warehouse pack copies some scripts under the %TWH_TOPDIR%/tools/bin directory, so you must ensure that they are always in line with the versions located under the %TWH_TOPDIR%/apps/amx/v511/misc directory.

Linking ETL steps

Scheduling ETL steps is something that has to be done with care. One ETL cannot start before the other is finished. For example, the AMY process cannot start before the AMX process is finished. This means that you have to monitor the AMX runtime and give it enough time to finish before you schedule the AMY process. Running the AMY process before the AMX process finishes will cause serious problems.

An easy way to get around this is to link the last step of the AMX process to the first step of the AMY process (or any other ETL2 process). We will use the AMX and AMY processes as an example. Complete the following steps:

1. From the Data Warehouse Center, expand the **AMX_IBM_TIVOLI_Monitoring_v5.1.1_Subject_Area** subject area, right-click the **AMX_c05_ETL1_Process**, and select **Task Flow** → **Steps**, as shown in Figure 8-5 on page 240.

Archived

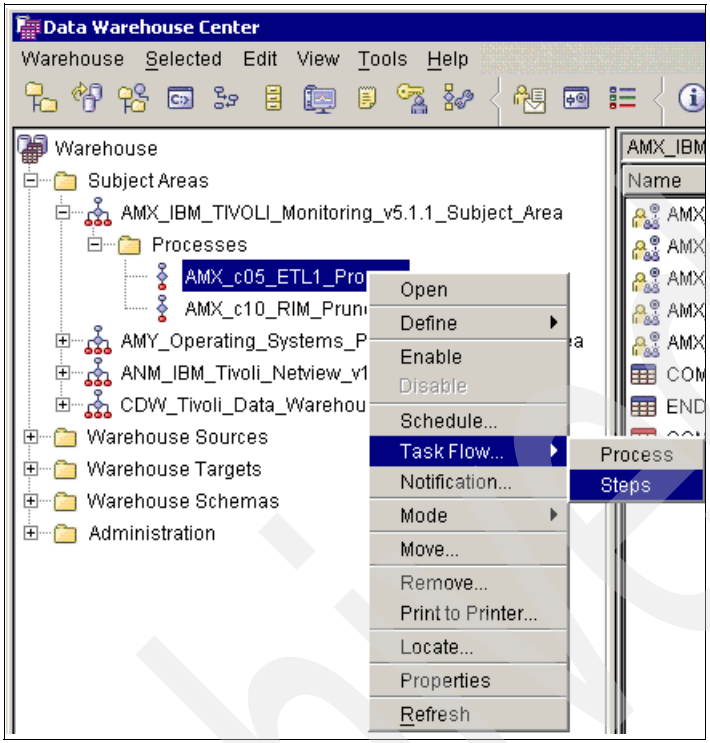


Figure 8-5 Data Warehouse Center Task Flow

2. In the Task Flow window, click the **Create Shortcut** button. Open the **AMY_Operating_Systems_PAC_v1.1.0_Subject Area** tree, locate the **AMY_c05_s010_Update** step, and add it by clicking the **“>”** button. Click **OK** to finish adding the shortcut.
3. Now, right-click in the window and select **Add**. This will add a line to the task flow, as shown in Figure 8-6 on page 241.

Archived

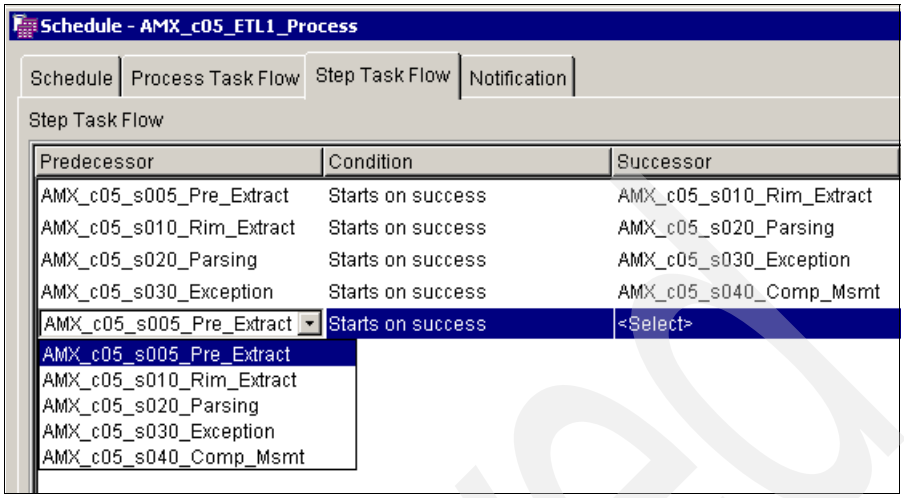


Figure 8-6 Add task flow step

4. Select the last step of the AMX run (AMX_c05_s040_Comp_Msmt) as the Predecessor and select the newly added AMY step as the Successor (AMY_c05_s010_Update). Leave the Condition "`Starts on success`".
5. Click **OK** to finish this step. In the main Data Warehouse window, you will see that the AMX_c05_s010_Update step was added as a link to the AMX process. Refer to Figure 8-3 on page 232 to see what this link looks like.

Archived

Archived

9

IBM DB2 UDB administration for Tivoli Data Warehouse

This chapter describes the necessary steps to administer IBM DB2 Universal Database Enterprise Edition (referred to as DB2) in the Tivoli Data Warehouse environment. As you know, DB2 is the base for Tivoli Data Warehouse. When you install the control server or install a warehouse pack, in effect, you are configuring the DB2 Data Warehouse Manager.

In addition, all the central data warehouse and data mart databases are DB2 databases. Therefore, to install and manage a Tivoli Data Warehouse environment, you need a basic knowledge of DB2 and the DB2 Warehouse Manager.

In this chapter, we discuss the following topics:

- DB2 general concepts
- Installing and configuring the DB2 client and server
- Installing IBM DB2 Warehouse Manager
- Basic database administration
- Basic DB2 Data Warehouse administration
- Configuring DB2 to work with Tivoli Data Warehouse V1.3

© Copyright IBM Corp. 2005. All rights reserved.

9.1 DB2 general concepts

In this section, we start by presenting important concepts about DB2. We assume that you are familiar with relational database and SQL language concepts. We compare the DB2 terminology and concepts with Oracle terminology and concepts, so if you have Oracle background, this section will be especially useful to help you understand the “DB2 side of the things” required for Tivoli Data Warehouse administration.

The following list defines some key terms:

- Instance

  DB2, similar to Oracle, has the concept of an instance. An instance is the context where DB2 command and functions are executed. You can have one or more instances in a server. Each instance has a different space address. Instances are implemented as services (on Windows) or daemons (on UNIX).

- Database

  From a user’s perspective, a database is set of related tables, but from a DBA’s perspective, a database includes all the other objects necessary to hold and manage these tables. Examples of objects are indexes, views, user, groups, and so on.

  Different from Oracle, a DB2 instance can hold more than one database. The databases in one instance are independent, but they use the same space address.

  Figure 9-1 on page 245 shows an example. The left side of figure shows two system (or servers). The server ISTANBUL has one instance (db2tdw) that has three databases (TWH_CDW, TWH_MART, and TWH_MD). We also see in the figure that a database can have many objects. The right side of figure shows several tables of the TWH_MD database.

Archived

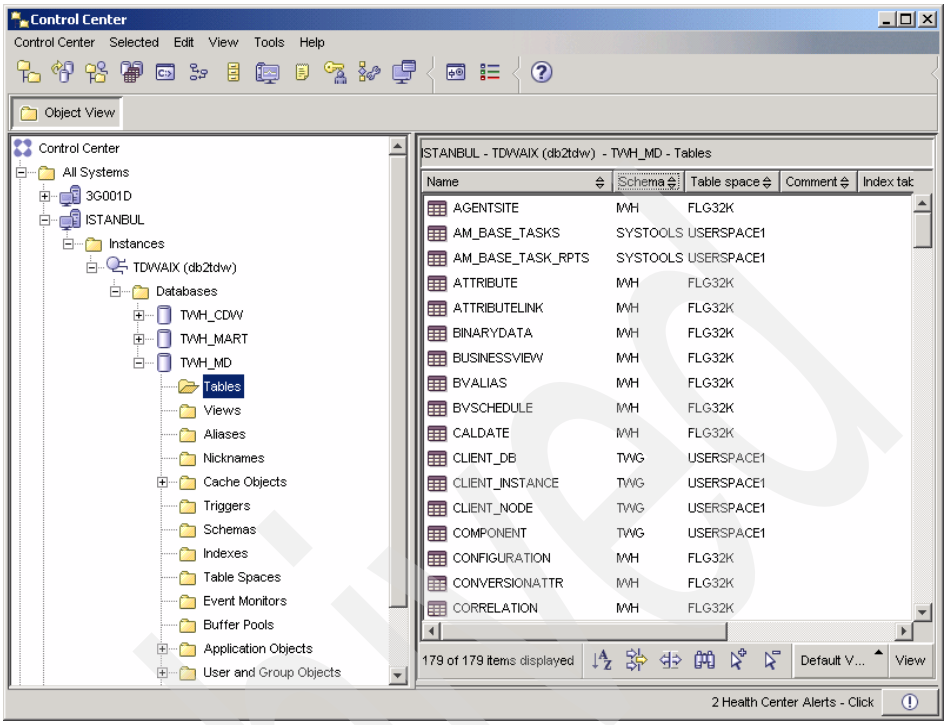


Figure 9-1 Instances, databases, and tables overview

- Schema

  A schema is a collection of named objects. A schema provides a logical classification of objects in the database. Tables, views, indexes, triggers are some of the objects that a schema can contain.

  In DB2, a schema is not associated with an user, which is different from Oracle. In Oracle, each user is also a schema, with the same name in the database.

- Tablespace

  A tablespace is a logical portion of database in which physical files for database tables can be stored. The tablespace concept here is the same as the Oracle database.

- Directory

  In DB2, directories are binary files that store information about the databases and their connectivity.

Archived

There are four main directories:

- System database directory (or system DB directory)

  The system database directory contains information about all the databases you can connect to from your DB2 system.

- Local database directory (or local DB directory)

  The local database directory contains information about local databases, that is, the databases residing in the machine in which you are currently working.

- Node directory

  The node directory is used to store all connectivity information for remote databases.

- Database Connection Services (DCS) directory

  The DCS directory contains connectivity information for host databases residing on an IBM @server® zSeries or iSeries™ machine.

- Logs

  Logs are used to record the operations that change the database. Logs are essential to recover the database in case of an instance crash or a media failure. This concept is the same as the REDO logs in Oracle.

- Buffer pools

  A buffer pool is an area in memory reserved to act as cache for pages (or block in Oracle nomenclature). All table and index pages go to this cache in order to be processed. This cache minimizes access to the disk, trying to hold in memory the most accessed pages.

  The main difference between Oracle and DB2 is that Oracle has one buffer pool (it can be divided in default, keep, and recycle pools). DB2 can have many buffer pools, and each tablespace is associated with one. This enables DB2 to work with different page sizes at same time.

9.2 Installing and configuring the DB2 client and server

As described in 3.1.3, “Database requirements” on page 43, Tivoli Data Warehouse V1.3 is dependent on IBM DB2 Universal Database Enterprise Edition server for a single system installation, or for the control server, central data warehouse, and data mart components of a distributed installation. For Crystal Enterprise Professional for Tivoli in a distributed Tivoli Data Warehouse installation, you can use the IBM DB2 client at the proper level to access the data mart databases.

This section provides information about the following topics:

- Installing the DB2 server on Windows
- Installing the DB2 server on AIX
- Installing the DB2 client

9.2.1 Installing the DB2 server on Windows

This section describes the IBM DB2 Universal Database Enterprise Edition Version 8.2 server installation process on Microsoft Windows.

> **Note:** Use the installation media provided with the Tivoli Data Warehouse product. This ensures that you install the correct version and fix pack of DB2.

To install the DB2 server on Windows, complete the following steps:

1. Insert the DB2 installation media into the CD-ROM drive. If the program does not start automatically when you insert the DB2 installation CD, run the `setup.exe` program in the root directory of the CD.
2. Click **Install Products**.
3. Select the **DB2 UDB Enterprise Server Edition** product. Click **Next**.
4. In Welcome to the DB2 Setup wizard window, click **Next**.
5. Accept the terms of the license agreement, and click **Next**.

Archived

6. In Select the installation type window, select the **Typical** installation. Also select the **Data warehousing** and **Satellite administration capability** functions. See Figure 9-2. Click **Next**.

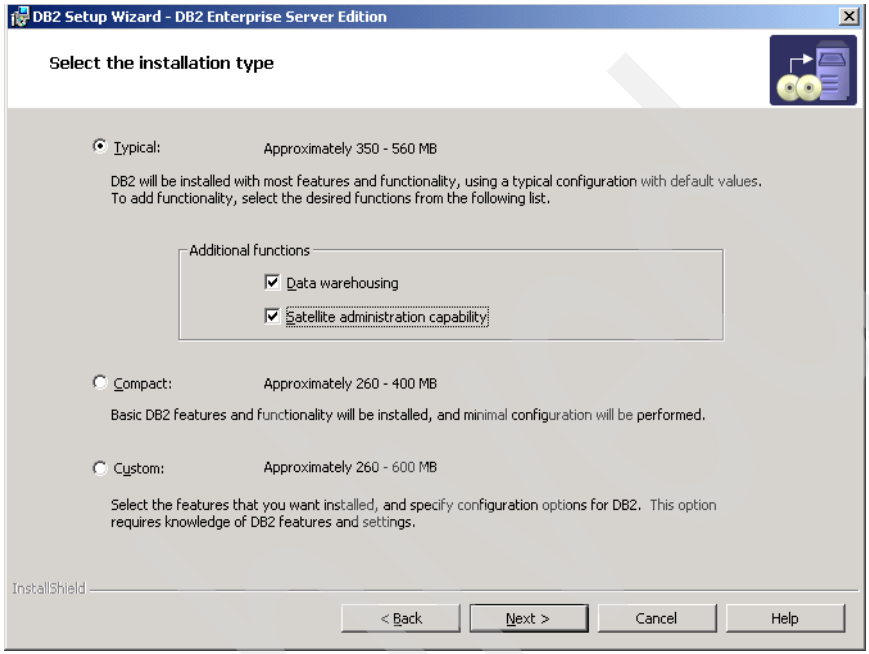


Figure 9-2 Select the installation type window

7. If you receive a warning about the firewalls, click **OK** after reading it.
8. In the Select the installation action window, select to install DB2 UDB Enterprise Server Edition on your system. Click **Next**.
9. Select the installation folder. Click **Next**.

> **Tip:** Install DB2 in a directory that does not contain spaces in its name to avoid potential problems.

10. Enter your DB2 user name and password. We selected and set the password for the db2admin user ID. Select the **Use the same user name and password for the remaining DB2 services** option, as shown in Figure 9-3 on page 249. Click **Next**.

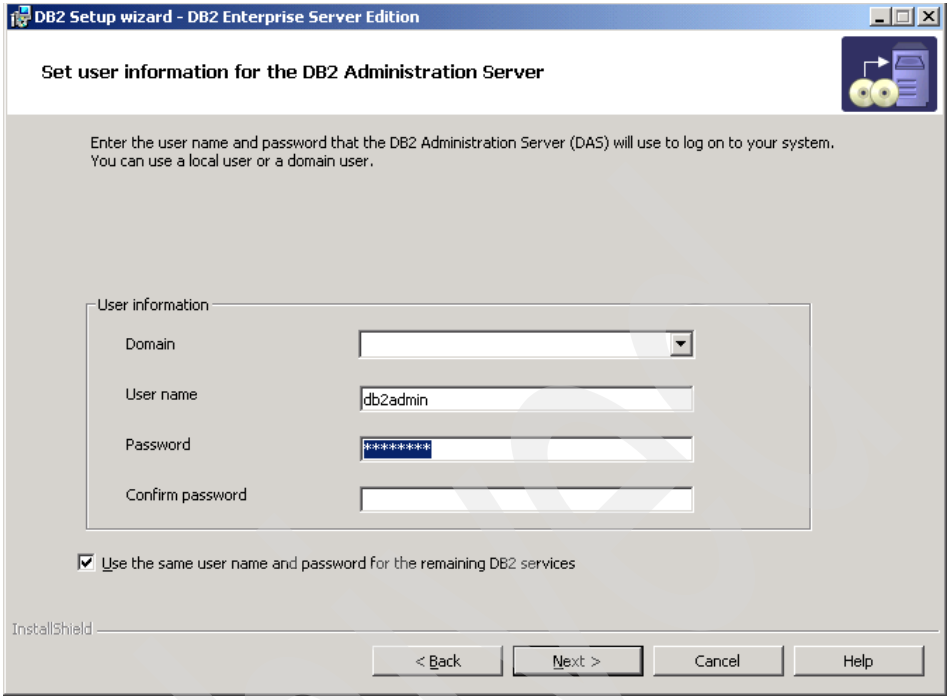


Figure 9-3 Set user information for the DB2 Administration Server window

> **Tip:** When you specify a user that does not already exist on your system, the DB2 installation wizard creates the user for you. It ensures that the user has the correct roles and privileges. If the user already exists, consider deleting it and letting the DB2 installation re-create it.

11. Set up the administration contact list, as appropriate for your system, and click **Next**. If you receive a warning about your notification SMTP server, click **OK** after reading the warning.
12. In the Create a DB2 instance window, select **Create the Db2 instance**. Click **Next**.
13. In the Configure DB2 Instances window, accept the defaults. Click **Next**.
14. In the Select the metadata to prepare window, select **Prepare the warehouse control database**, and click **Next**.
15. In the Specify a local warehouse control database, accept the default information. Click **Next**.
16. Specify a contact for health monitor information, and click **Next**.

17. Click **Install** to start the installation.
18. Click **Finish** in the completion window, as shown in Figure 9-4.

Figure 9-4 Installation complete window

9.2.2 Installing the DB2 server on AIX

This section describes the IBM DB2 Universal Database Enterprise Edition Version 8.2 server installation process on AIX.

> **Note:** Use the installation media provided with the Tivoli Data Warehouse product. This ensures that you install the correct version and fix pack of DB2.

To install the DB2 server on AIX, complete the following steps:

1. Insert the DB2 installation media into the CD-ROM drive.
2. Mount the CD using the **`mount -r /dev/cd0 /cdrom`** command.
3. Copy the file /cdrom/ese.tar.Z into stage directory. Move into this stage directory.
4. Uncompress and untar this file using **`uncompress ese.tar.Z && tar -xvf ese.tar`**.

5. Run `./ese/db2setup`.
6. Click **Install Products**.
7. Select the **DB2 UDB Enterprise Server Edition** product. Click **Next**.
8. In Welcome to the DB2 Setup wizard, click **Next**.
9. Accept the terms of the license agreement, and click **Next**.
10. In the Select the installation type window, select the **Typical** installation. Also select the **Data warehousing** and **Satellite administration capability** functions, as shown in Figure 9-5. Click **Next**.

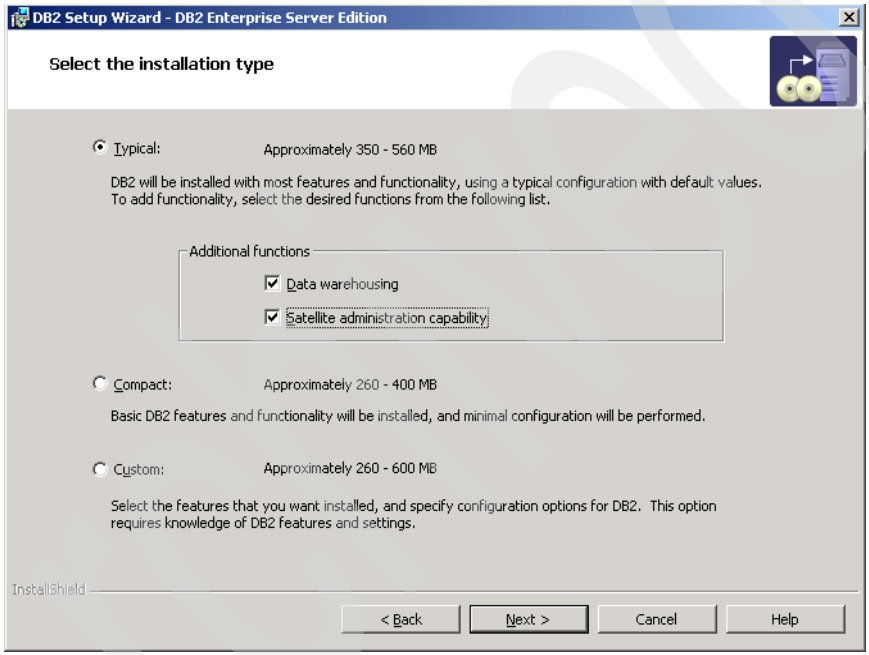


Figure 9-5 Select the installation type window

11. In the Select the installation action window, select to install DB2 UDB Enterprise Server Edition on your system. Click **Next**.

12. In the Set user information for the DB2 Administration Server window, enter and confirm the password for the user. This user is used to run the administration server. The DB2 installer creates this user. See Figure 9-6. Click **Next**.

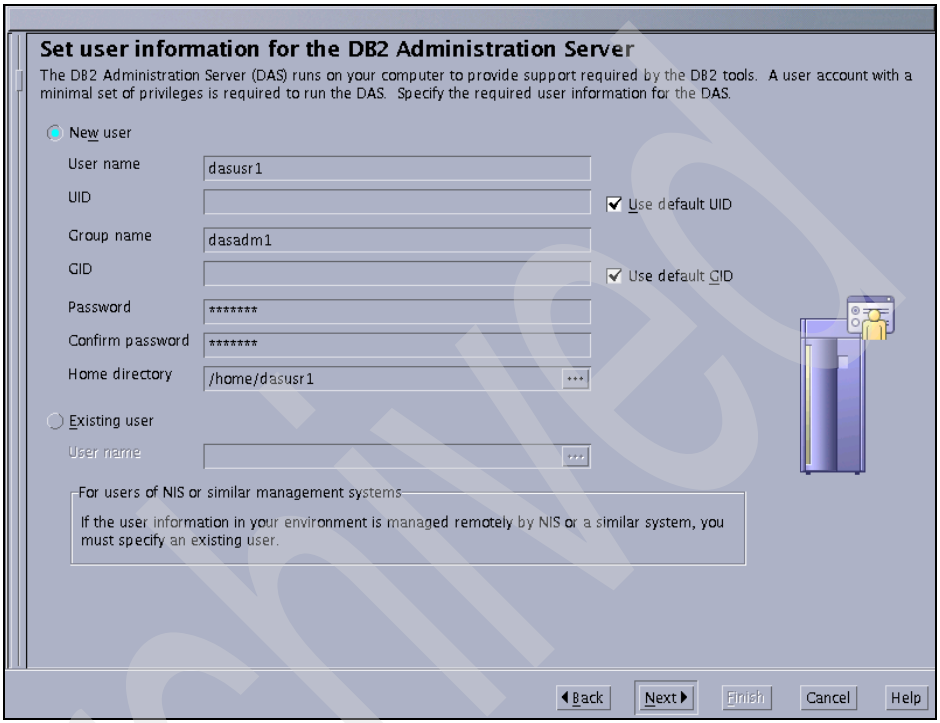


Figure 9-6 Set user information for the DB2 Administration Server window

Archived

13. In the Set up a DB2 instance window, select **Create a DB2 instance - 32 bit**, as shown in Figure 9-7. Click **Next**.

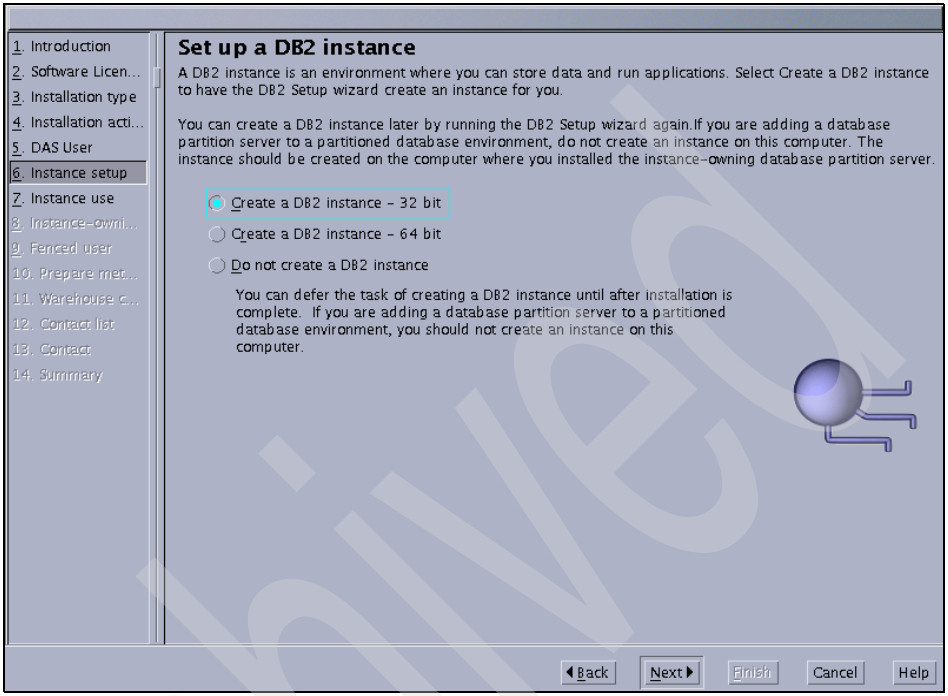


Figure 9-7 Set up a DB2 instance window

Archived

14. In the Select how the instance will be used window, select **Single-partition instance**, as shown in Figure 9-8. Click **Next**.

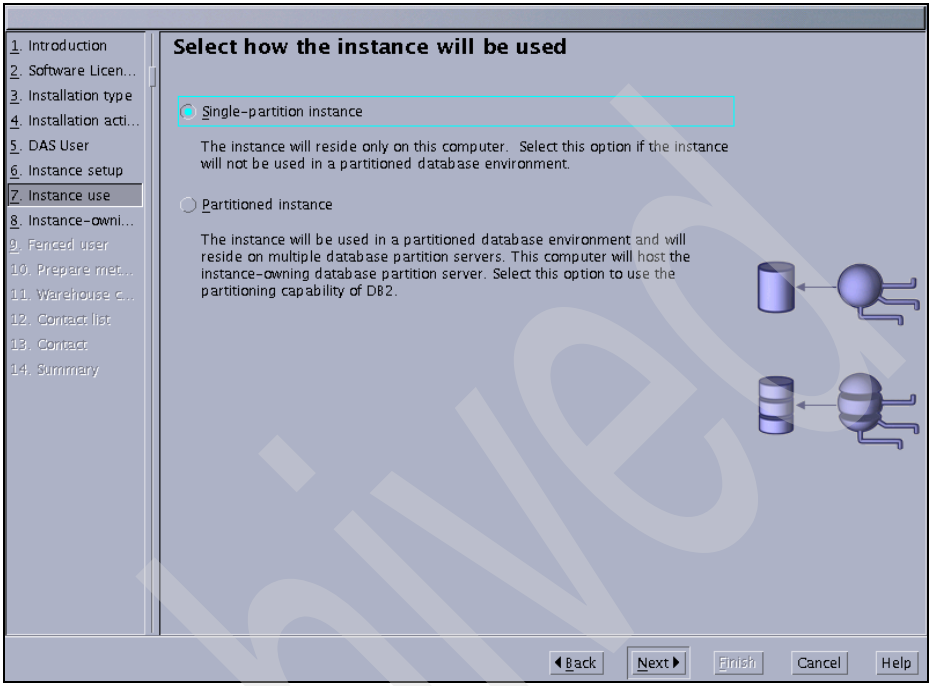


Figure 9-8 Select how the instance will be used window

Archived

15. In the Set user information for the DB2 instance owner window, enter and confirm the password for the user. This user is used to hold the instance. The DB2 installer creates this user. See Figure 9-9.

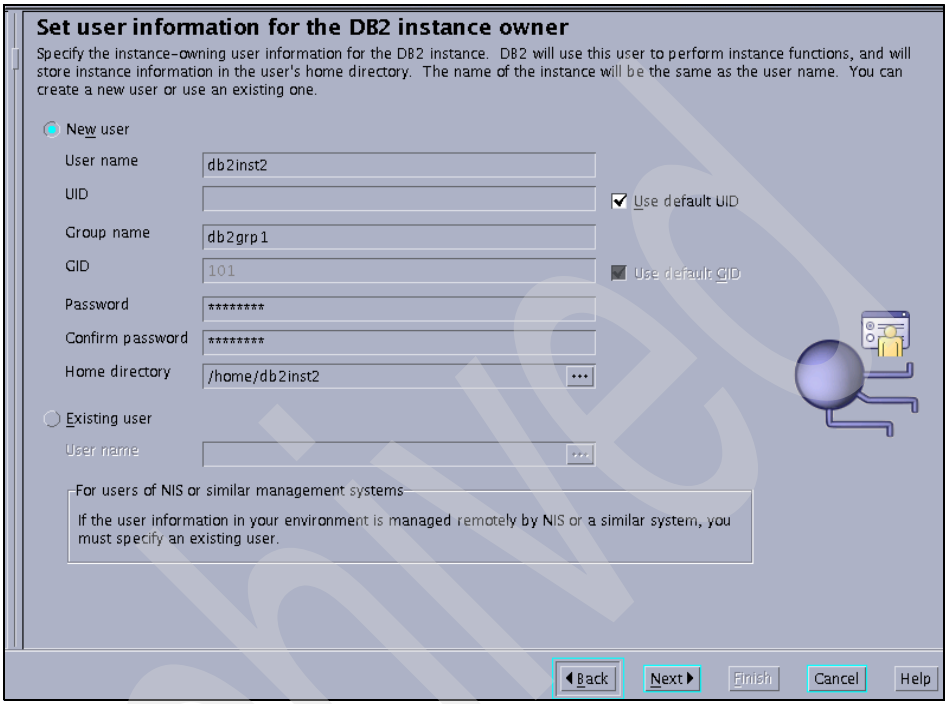


Figure 9-9 Set user information for the DB2 instance owner window

Archived

16. In Set user information for the fenced user window, enter and confirm the password for the user. This user is used to run user-defined functions (UDFs) and stored procedures. For security reasons, you should not use the instance owner user. The DB2 installer creates this user. See Figure 9-10. Click **Next**.

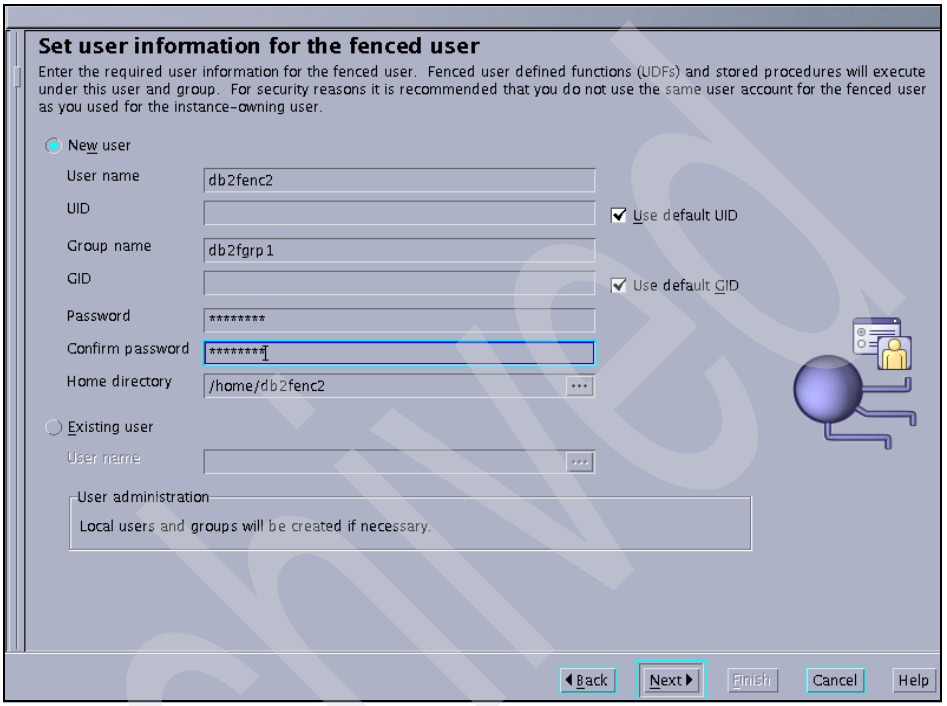


Figure 9-10 Set user information for the fenced user window

17. In the Select the metadata to prepare window, select **Prepare the warehouse control database**, and click **Next**.

18. Enter your DB2 instance owner user name and password. Select the **Use the same user name and password for the remaining DB2 services** option. Click **Next**.

19. Set up the administration contact list, as appropriate for your system, and click **Next**. If you receive a warning about your notification SMTP server, click **OK** after reading it.

20. Specify a contact for health monitor information, and click **Next**.

21. Click **Install** to start the installation.

Archived

22. Click **Finish** in the completion window, as shown in Figure 9-11.

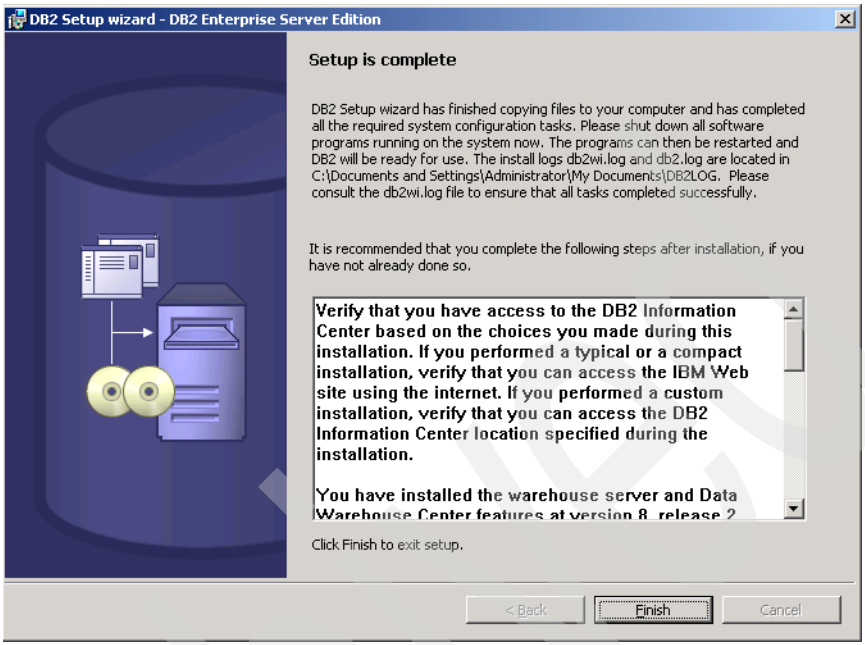


Figure 9-11 Installation complete window

9.2.3 Installing the DB2 client

This section describes the installation of the IBM DB2 client on Windows.

> **Note:** The server of IBM DB2 Universal Database Enterprise Edition has a client included.

If you are using an existing installation of the IBM DB2 client, ensure that it is at Version 8.2.

The IBM DB2 client software can be obtained by installing the IBM DB2 Administration client package that comes with the IBM DB2 installation media provided by Tivoli Data Warehouse V1.3. Use the IBM DB2 installation media provided with the Tivoli Data Warehouse to ensure that you get the correct version.

The IBM DB2 Administration client package can be installed by performing the following actions:

1. Insert the IBM DB2 installation media into the CD-ROM drive. If the program does not start automatically when you insert the DB2 installation CD, run the `setup.exe` program in the root directory of the CD. Click **Install**.
2. Select **DB2 Administration Client**. Make sure that the **DB2 Enterprise Edition** product is not selected.
3. Select the **Typical** installation.
4. You can change the default destination folder (optional). However, we used `c:\db2\sqllib` in our installation. Accept the default values for the remaining items in this window.
5. For the control center server user name, specify a user that does not already exist on your system. When the DB2 installation wizard creates the user for you, it ensures that the user has the correct roles and privileges. If the user already exists, consider deleting it and letting the IBM DB2 installation re-create it. For more information about DB2 naming rules, refer to *IBM DB2 UDB Quick Beginnings for Windows*, GC09-2971.

   In a safe place, record the IBM DB2 user name and password that was specified for the IBM DB2 client.
6. On the Start Copying Files window, click **Next**. Then, click **Finish** to complete the setup.

9.3 Installing IBM DB2 Warehouse Manager

In this section, we provide the steps required for installing IBM DB2 Warehouse Manager on Windows. The installation process is described in *Installing and Configuring Tivoli Data Warehouse*, GC32-0744.

> **Tip:** Use the IBM DB2 Warehouse Manager installation media provided with Tivoli Data Warehouse V1.3. This ensures that you install the correct version and fix pack level.

9.3.1 Installing IBM DB2 Warehouse Manager on a Windows platform

To install IBM DB2 Warehouse Manager on a Windows platform, complete the following steps:

1. Stop *all* IBM DB2 processes before proceeding with the installation. Open a DB2 command window and issue the following commands:

   ```
   db2stop force
   db2admin stop
   ```

   Also, on the Windows Services panel, stop the following services:

   – DB2-DB2CTLSV
   – DB2 JDBC Applet Server
   – DB2 License Server
   – DB2 Security Server
   – DB2 Warehouse server
   – DB2 Warehouse logger

2. Load the IBM DB2 Warehouse Manager installation media.
3. Select **Start** → **Run**. Type in `D:\setup.exe`, and click **OK** to start the installation. From the Installation window, select **Install Products**.
4. The Select Products window opens. Make sure that **DB2 Warehouse Manager** is selected. Click **Next**.
5. In the Welcome window, click **Next**.
6. Accept the terms of the license agreement, and click **Next**.
7. In the Select the installation action window, select to install DB2 Warehouse Manager on your system. Click **Next**.
8. In the Select the features you want to install window, make sure that only **Warehouse Agent** and **Documentation** are selected, as shown in Figure 9-12 on page 260. Click **Next**.

Archived

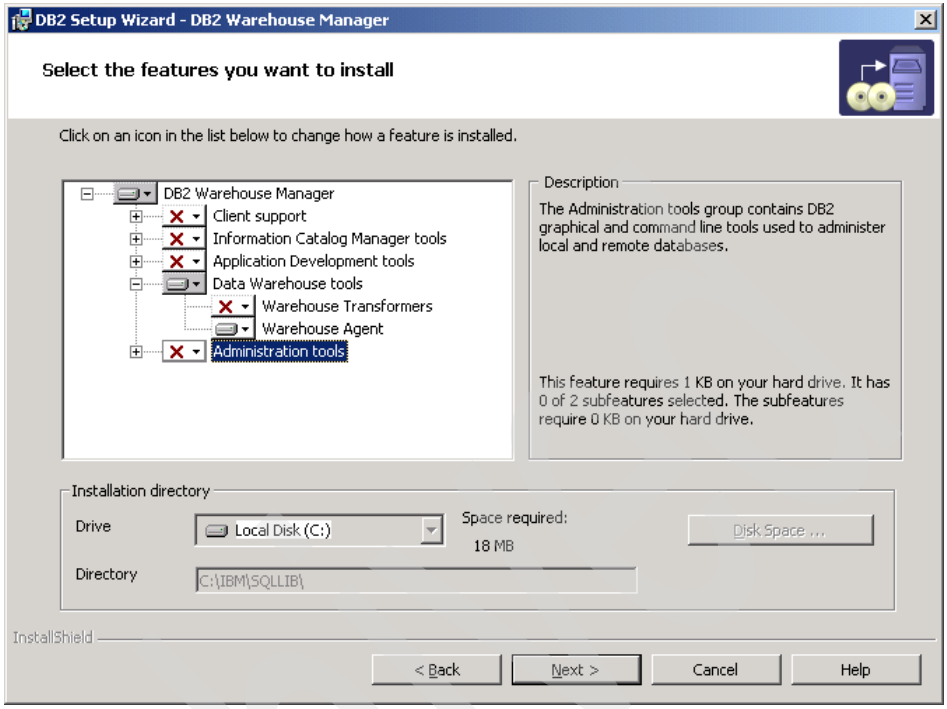


Figure 9-12 Select the DB2 Warehouse Manager components

9. If you receive a warning about firewalls, click **OK** after reading it.
10. In Select the languages to install window, select the languages you want to install. These languages are used for the documentation and messages. Click **Next**.
11. At the Start Copying Files window, you can review the installation options. Click **Next**.
12. When the installation process is complete, click **Finish**.
13. Reboot the machine.

9.3.2 Installing IBM DB2 Warehouse Manager on an AIX platform

To install IBM DB2 Warehouse Manager on an AIX platform, complete the following steps:

1. Load the IBM DB2 Warehouse Manager installation media.

2. Mount the CD. In our case, the CD-ROM is device cd0, and we mount it in the /cdrom directory. Therefore, in our case, we issue the **`mount -r /dev/cd0 /cdrom`** command.
3. Change to directory where you mounted the CD-ROM. In our case, we issue **`cd /cdrom`**.
4. Run **`./db2setup`**.
5. Click **Install Products**. The Select Products window opens. Make sure that **DB2 Warehouse Manager** is selected. Click **Next**.
6. In Welcome window, click **Next**.
7. Accept the terms of the license agreement, and click **Next**.
8. In the Select the installation action window, select to install DB2 Warehouse Manager on your system. Click **Next**.
9. On the Select the features to install window, make sure that **Warehouse Agent** and are selected, as shown in Figure 9-13. Click **Next**.

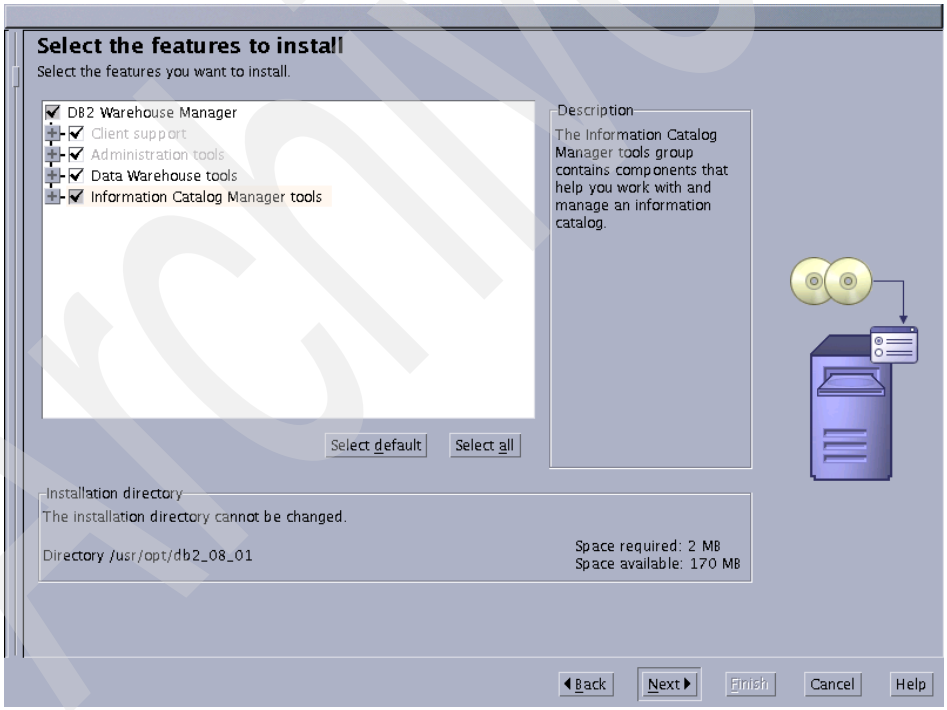


Figure 9-13 Select the features to install window

Archived

10. In Select the languages to install window, select the languages you want to install. These languages are used for the documentation and messages. Click **Next**.
11. At the Start Copying Files window, you can review the installation options. Click **Next**.
12. When the installation process completes, click **Finish**.

9.4 Basic database administration

In this section, we cover basic administration activities in a DB2 database. This is vast subject, and here, we only cover the more usual tasks and tools used in day-to-day administration. You can find more information in the DB2 documentation and help files.

9.4.1 DB2 tools

We can divide the DB2 tools in two groups:

- Command line tools
  - DB2 command window

    The DB2 command window is necessary in some environments to use the operating system command line environment to execute DB2 commands. For example, in Windows, if you open a command window and try to execute the **`db2 list database directory`** command, you receive a message showing `DB21061E Command line environment not initialized`. Now, if you execute **`db2cmd`**, or from the Windows Start menu, select **Programs** → **IBM DB2** → **Command Line Tools** → **Command Window**, another window opens. In that window, you can execute the above command. This command shows the cataloged databases. In a command window, all DB2 commands start with the word **`db2`**. You can also issue any operational system command.

    This tool can be difficult to the novice user, because you must know the command syntax. Nevertheless, it is very useful, because you can write batch files to automate repetitive tasks.

    In a UNIX machine, the environment for execution is set in the profile of the user. Therefore, the **`db2cmd`** command is not used. You can run DB2 commands if the user is configured to access DB2 in that user's profile.
  - DB2 command line processor

    The DB2 command line processor is an application that you use to execute DB2 commands and operating system commands.

This tool can be difficult to the novice, because you must know the command syntax. However, this tool is very useful, because you can write batch files to automate repetitive tasks.

- Graphical user interface (GUI)

  IBM DB2 Universal Database Enterprise Edition provides a whole array of graphical user interfaces (GUIs), such as:

  – DB2 Command Center

    The Command Center is the main GUI you use to administer DB2. See Figure 9-14. Using this tool, you can, for example, create or catalog an instance or database and start and stop instance. In addition, you can launch the other GUI tools from the Command Center.

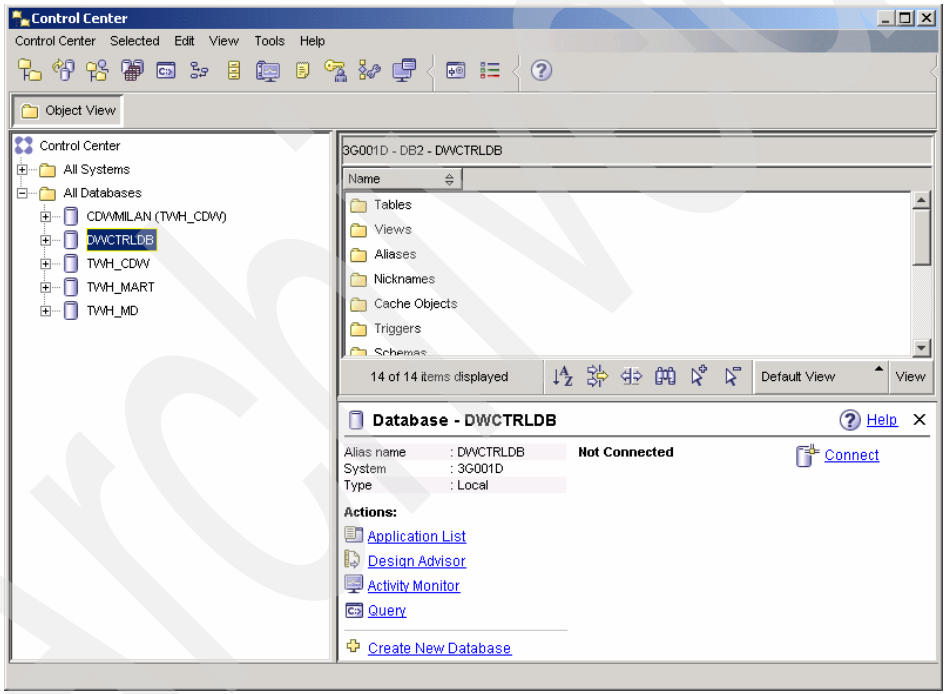


Figure 9-14 Command Center window

  – DB2 Health Center

    The DB2 Health Center is used to monitor the state of the DB2 environment. You can start the DB2 Health Center from any DB2 tool launchpad, or by entering the **db2hc** command at a command prompt.

    When you use DB2, a monitor continuously keeps track of a set of health indicators. If the current value of a health indicator is outside the

acceptable operating range defined by its warning and alarm thresholds, the Health Monitor generates a health alert. DB2 comes with a set of predefined threshold values, which you can customize.

– DB2 Task Center

DB2 Task Center is used to run tasks, either immediately or according to a schedule, and to notify people about the status of completed tasks. See Figure 9-15. You can start the DB2 Task Center is used to run tasks, either immediately or according to a schedule, and to from a DB2 tool launchpad, or by entering the **db2tc** command from a command prompt.

A task is a script accompanied by associated failure or success conditions, schedules, and notifications.

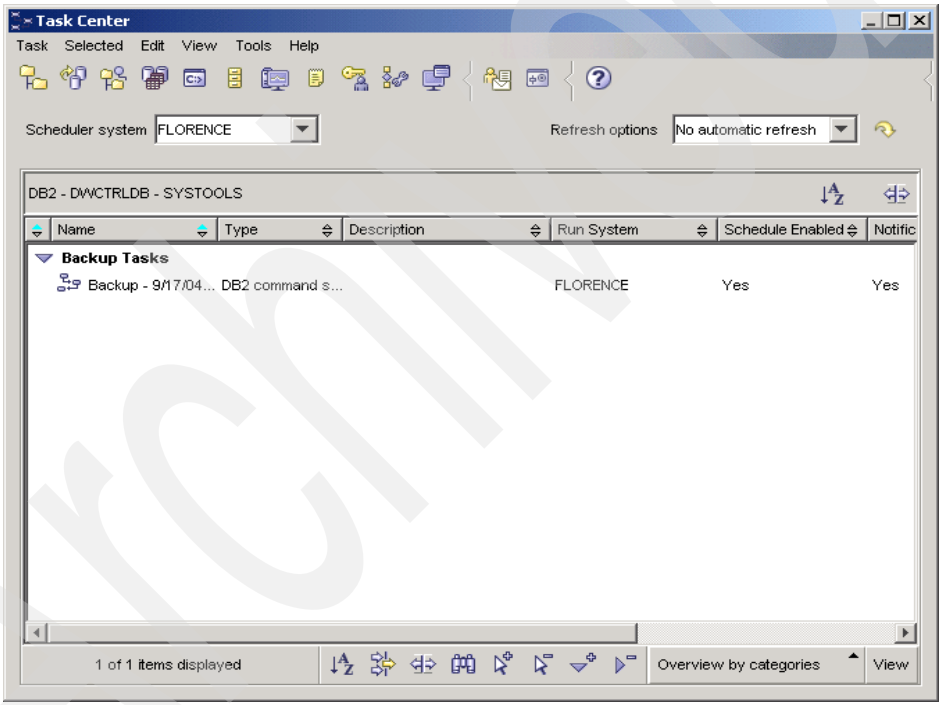


Figure 9-15 DB2 Task Center window

9.4.2 Basic activities and commands

We show here how to execute common tasks that you need to administer a Tivoli Data Warehouse environment. We do not discuss how to create a instance or database, because you do not need to do this manually in this environment. The instances are created when you install DB2, and the database is created through

the Tivoli installer. We describe here only the line commands. However, these activities can be executed using the DB2 Command Center GUI.

- Starting and stop an instance

  To stop an instance, in a command window, type:

  ```
  db2stop or db2stop force
  ```

  To start an instance, type:

  ```
  db2start
  ```

- Starting and stopping the administration server

  To stop the administration server, type:

  ```
  db2adm stop
  ```

  To start the administration server, type:

  ```
  db2adm start
  ```

- Listing, migrating, and updating a DB2 instance

  To list the existing databases on a server, type:

  ```
  db2list
  ```

  Instance migration is necessary when you install a newer version of the database or when you are migrating from a 32-bit to a 64-bit instance. In Windows, you do not need to do this manually. This is done implicitly during the migration process. In UNIX, use:

  db2imigr *instance_name*

  When you install a fix pack, you need to update the instances. Use:

  db2ipudt *instance_name*

- Setting up instance parameters

  It is necessary to adjust some parameters in DB2 to use Tivoli Data Warehouse. We show you how to browse the actual values and to how to change them. These changes are recognized by the instance after a restart.

  To see the actual values of a variable, type:

  db2set *variable-name*

  To change a variable, type:

  db2set *variable-name=new-value*

- Verifying the installed version and fix pack level

  To verify the version and currently installed fix pack, issue the command:

  ```
  db2level
  ```

Archived

9.5 Basic DB2 Data Warehouse administration

In this section, we cover the basic administration activities in DB2 Data Warehouse. We only cover the activities used in day-to-day administration for a Tivoli Data Warehouse environment. You can refer to the DB2 documentation for more information. Figure 9-16 shows the initial window of the Data Warehouse Center.

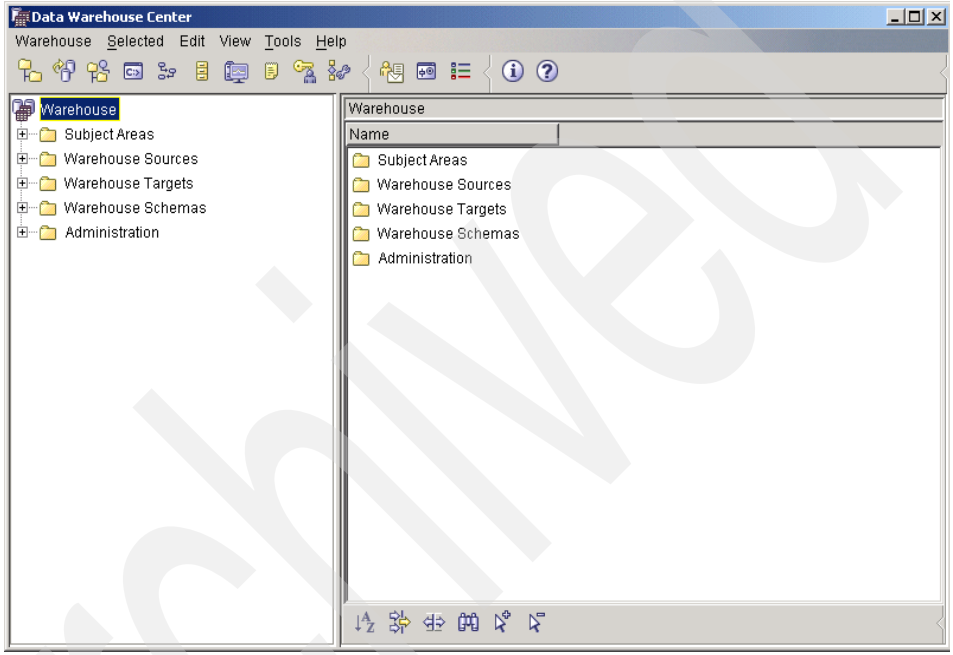


Figure 9-16 Data Warehouse Center window

Each warehouse pack you install appears as different subject in the Data Warehouse Center. Each subject has one or more processes. You must schedule and manage these processes in theTivoli Data Warehouse environment. See Figure 9-17 on page 267.

Archived

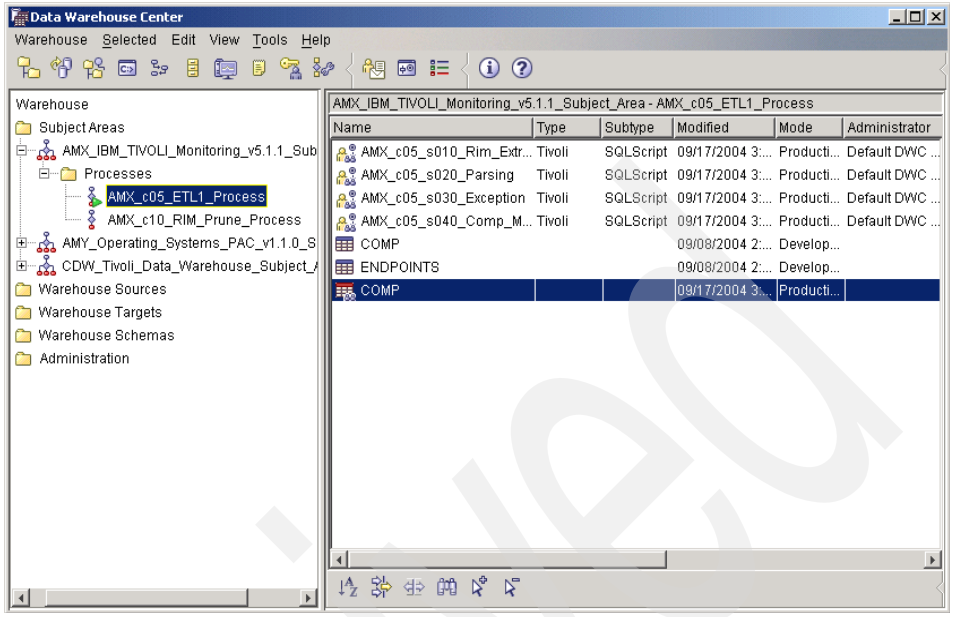


Figure 9-17 Data Warehouse Center processes window

Each process can be in a Production, Test, or Development mode. You only can change a process step when it is in the Development mode. In addition, you can only schedule a process when it is the Production mode. The Test mode, as the name implies, must be used to test your process before it goes into Production mode. To change the status of a process, right-click the process in the Data Warehouse Center window, and then select the status that you want. See Figure 9-18 on page 268.

Archived

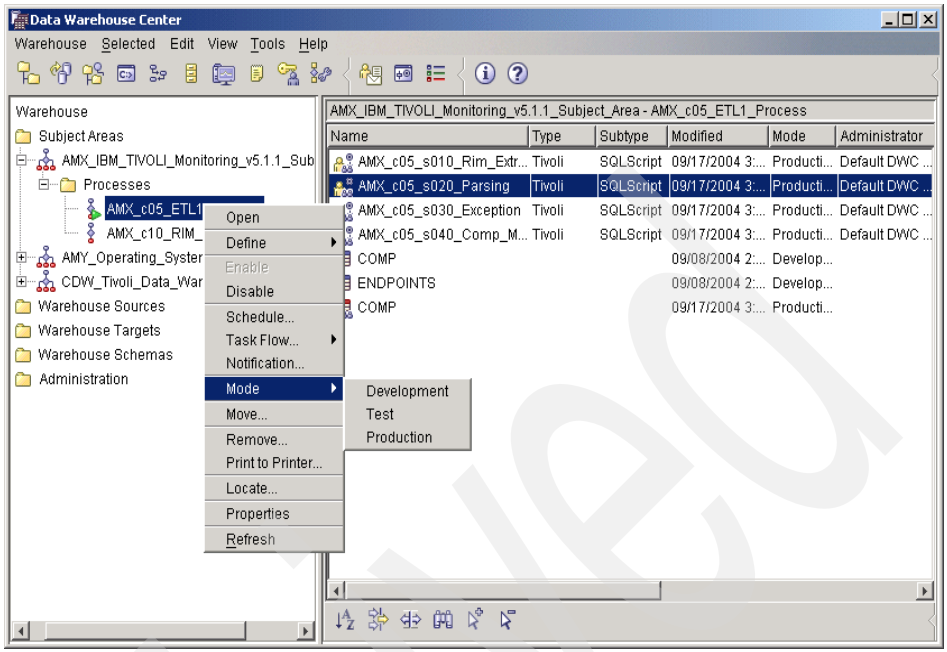


Figure 9-18 Changing the mode of a process

Next, you need to schedule its execution. To do this, right-click the process you want to schedule in the Data Warehouse Center window, and select **Schedule**. In the Process schedule window, complete your schedule for this process, and click **Add**, and then **OK**. See Figure 9-19 on page 269.

Archived

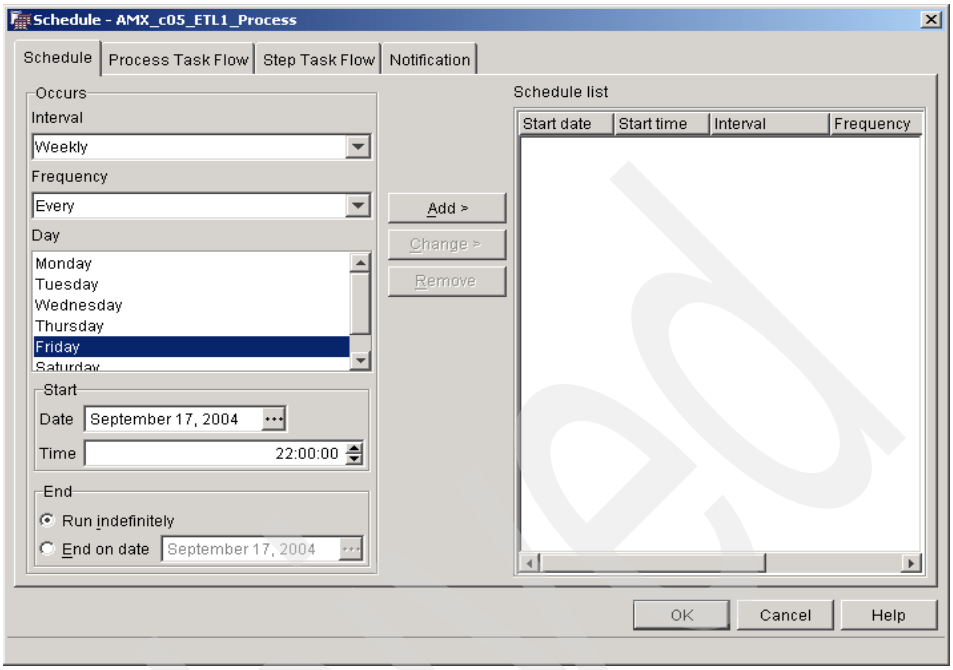


Figure 9-19 Process schedule setup window

9.6 Configuring DB2 to work with Tivoli Data Warehouse V1.3

This section will help a DB2 DBA to know what is necessary from the DB2 point of view for Tivoli Data Warehouse V1.3. Consider the following items:

- DB2 version

  All the databases of Tivoli Data Warehouse V1.3, the data mart databases, central data warehouse database, and control server must be at least at Version 8.2. Note that Version 8.2 is also known as Version 8.1 Fix Pack 7.

- Components to install

  On UNIX systems, make sure that the administration client is installed. If you can start the DB2 Control Center, the client is installed.

  On Windows systems, make sure that you install the following components when installing DB2: Data warehousing and Satellite administration capability.

Archived

- Naming convention

  It is important you make sure that other applications using the DB2 instance do not have database names that are same as Tivoli Data Warehouse. Table 9-1 lists the database names used by Tivoli Data Warehouse. The databases are created based on these naming conventions.

Table 9-1 Tivoli Data Warehouse database naming conventions

Database	ODBC client/server connection name	Database alias	Database name
Control database on the control server.	TWH_MD	TWH_MD	TWH_MD
First DB2 UDB central data warehouse on a Windows or UNIX system.	TWH_CDW	TWH_CDW	TWH_CDW
First DB2 UDB data mart on a Windows or UNIX system.	TWH_MART	TWH_MART	TWH_MART
Subsequent DB2 UDB central data warehouses, including those on z/OS systems.	TWH_CDWn, where n, starting with 1, indicates the order in which the central data warehouse was created.	TCDWn, where n matches the number in TWH_CDWn.	TCDWn, where n, starting with 1, indicates the order in which the data mart was created.
Subsequent DB2 Universal Database data marts, including those on z/OS systems.	TWH_MARTn, where n, starting with 1, indicates the order in which the data mart was created.	TMARTn, where n matches the number in TWH_MARTn.	TMARTn, where n, starting with 1, indicates the order in which the data mart was created.

  Tivoli Data Warehouse also creates node aliases named TDWn and local DCS databases named TDCSn. The local DCS database is created when you install a central data warehouse or data mart on a z/OS system.

- DB2 parameters

  Tivoli Data Warehouse V1.3 requires some specific values for DB2 parameters. These parameters must be configured as follows:

  APP_CTL_HEAP_SZ The application control heap size for all central data warehouse must be set to 512 or higher.

  DB2_ENABLE_LDAP This must be set to NO in all instances.

  DB2CODEPAGE This must be set to 1208 in all instances.

Archived

A

Additional material

This Redbook refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this Redbook is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

ftp://www.redbooks.ibm.com/redbooks/SG246343

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select **Additional materials** and open the directory that corresponds with the Redbook form number, SG246343.

Using the Web material

The additional Web material that accompanies this IBM Redbook includes the following file:

File name	*Description*
SG246343.zip	Sizing spreadsheet

© Copyright IBM Corp. 2005. All rights reserved.

System requirements for downloading the Web material

The following system configuration is recommended:

Hard disk space: 1 MB minimum
Operating system: Microsoft Windows/Linux

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material ZIP file into this folder.

Archived

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this Redbook.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 274. Note that some of the documents referenced here might be available in softcopy only.

- *DB2 Warehouse Management: High Availability and Problem Determination Guide*, SG24-6544
- *IBM @server Certification Study Guide - AIX 5L Performance and System Tuning*, SG24-6184
- *Introducing IBM Tivoli Monitoring for Web Infrastructure*, SG24-6618
- *Tivoli Data Warehouse 1.2 and BusinessObjects*, REDP-9116

Other publications

These publications are also relevant as further information sources:

- *Enabling an Application for Tivoli Data Warehouse*, GC32-0745
- *IBM DB2 UDB Quick Beginnings for DB2 Servers V8.2*, GC09-4836
- *IBM DB2 UDB Quick Beginnings for Windows*, GC09-2971
- *IBM DB2 Warehouse Manager Standard Edition Installation Guide*, GC27-1122
- *IBM Tivoli Monitoring Resource Model Reference*, SH19-4570
- *IBM Tivoli Monitoring User’s Guide*, SH19-4569
- *IBM Tivoli NetView Warehouse Enablement Pack: Implementation Guide*, SC32-1237
- *Installing and Configuring Tivoli Data Warehouse*, GC32-0744
- *Tivoli Data Warehouse Release Notes*, SC32-1399

© Copyright IBM Corp. 2005. All rights reserved.

Online resources

These Web sites and URLs are also relevant as further information sources:

- Tivoli Data Warehouse Support Web site

  http://www.ibm.com/software/sysmgmt/products/support/TivoliDataWarehouse.html

- DB2 documentation

  http://www.ibm.com/software/data/db2/udb/support/manualsv8.html

- DB2 developerWorks article “Best practices for tuning DB2 UDB v8.1 and its databases: A handbook for high performance”

  http://www.ibm.com/developerworks/db2/library/techarticle/dm-0404mcarthur/index.html

- DB2 developerWorks article “Improve database performance on file system containers in IBM DB2 UDB V8.2 using Concurrent I/O on AIX”

  http://www.ibm.com/developerworks/db2/library/techarticle/dm-0408lee/index.html

- IBM Cloudscape™ resources and documentation

  http://www.ibm.com/developerworks/db2/zones/cloudscape/
  http://www.ibm.com/software/data/cloudscape/version51/pubs

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Archived

Index

Archived

© Copyright IBM Corp. 2005. All rights reserved.

E

F

G

I

Archived

J

K

L

M

N

Archived

Archived

T

Archived

U

V

W

Z

Archived

Tivoli Data Warehouse Version 1.3: Planning and Implementation

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

Archived

Archived

Archived

Tivoli Data Warehouse Version 1.3: Planning and Implementation

A first look at Tivoli Data Warehouse Version 1.3

Planning, implementing, and troubleshooting

Performance optimization

With IBM Tivoli Data Warehouse, you can analyze historical trends from various Tivoli and customer applications. The Tivoli Data Warehouse infrastructure enables a set of extract, transform, and load (ETL) utilities to extract and move data from Tivoli application data stores to a central repository.

This IBM Redbook focuses on planning, installation, customization, use, maintenance, and troubleshooting topics related to the new features of Tivoli Data Warehouse Version 1.3. This is done using a number of case study scenarios and warehouse packs.

This book has a comprehensive chapter about how to optimize the overall performance of a Tivoli Data Warehouse implementation, which focuses on DB2 optimization techniques and remote warehouse agents.

Also included with the book is a sizing spreadsheet that will help you determine the database sizing requirements of Tivoli Data Warehouse and various warehouse packs.

Finally, this book provides handy information and tips about troubleshooting Tivoli Data Warehouse.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-6343-00 ISBN 0738491713