

IBM® DB2 Universal Database™



Installing and Administering a Satellite Environment

Version 8

IBM® DB2 Universal Database™



Installing and Administering a Satellite Environment

Version 8

Before using this information and the product it supports, be sure to read the general information under *Notices*.

This document contains proprietary information of IBM. It is provided under a license agreement and is protected by copyright law. The information contained in this publication does not include any product warranties, and any statements provided in this manual should not be interpreted as such.

You can order IBM publications online or through your local IBM representative.

- To order publications online, go to the IBM Publications Center at www.ibm.com/shop/publications/order
- To find your local IBM representative, go to the IBM Directory of Worldwide Contacts at www.ibm.com/planetwide

To order DB2 publications from DB2 Marketing and Sales in the United States or Canada, call 1-800-IBM-4YOU (426-4968).

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

© **Copyright International Business Machines Corporation 1999-2002. All rights reserved.**

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

Contents

About This Book	xi
Who Should Use This Book	xi
How This Book Is Structured	xi
Concepts	xiii
Satellite Environment	xiii
Satellite Control Server	xiv
Satellite Control Database	xv
Groups in a Satellite Environment	xvi
Satellites	xvii
Model Office and Its Role in a Satellite Environment	xviii
Application Versions and Batches in a Satellite Environment	xx
Satellite Synchronization	xxi
Administration of Groups of Satellites	xxiii
Satellite Administration Center	xxiv
Example of a Satellite Environment	xxv

Part 1. Installing and Migrating a Satellite Environment. 1

Chapter 1. Installing the Satellite Control Server and Satellites	3
Preparation for the Installation of a Satellite Environment	3
Installing and Setting Up the Satellite Control Server and Satellite Control Database	5
Disk Requirements for the Satellite Control Server	5
Software Requirements for the Satellite Control Server	5
Satellite Control Database Considerations	6
Setting Up the Satellite Control Server on AIX	7
Creating the DB2CTLSV Instance on AIX.	7
Creating the SATCTLDB Database on AIX	8
Customizing the SATCTLDB Database on Windows	9

Chapter 2. Installing DB2 Workgroup Server Edition and DB2 Enterprise Server Edition	11
---	-----------

Installing Single-Partition Database Environments on Windows-Based Platforms	12
Installation overview for DB2 servers (Windows)	12
Installing a DB2 server on Windows	14
Installation overview for DB2 servers (Windows)	15
Installation requirements for DB2 servers (Windows)	17
Memory requirements for DB2 servers (Windows)	19
Disk requirements for DB2 servers (Windows)	19
Extending the directory schema (Windows 2000 and Windows .NET)	20
User accounts required for installation of DB2 servers (Windows)	21
Starting the DB2 Setup wizard for a DB2 server installation (Windows)	23
Applying the latest FixPak	25
Verifying the installation using the command line processor (CLP).	26
Installing DB2 online documentation (Windows)	27
Installing Partitioned Database Environments on Windows-Based Platforms	29
Installing a partitioned DB2 server (Windows)	29
Installation overview for partitioned DB2 servers (Windows)	30
Installation requirements for a partitioned DB2 server (Windows)	33
Memory requirements for a partitioned DB2 server (Windows)	35
Disk requirements for a partitioned DB2 server (Windows)	36
Preparing the environment for a partitioned DB2 server (Windows)	37
Extending the directory schema (Windows 2000 and Windows .NET)	39
Installing the instance owning database partition server (Windows)	40
Verifying port range availability on participating computers	44

Installing database partition servers on participating computers (Windows)	45	Installing database partition servers on participating computers using a response file (UNIX)	86
Applying the latest FixPak	48	Updating the node configuration file (UNIX).	87
Verifying a partition database server installation (Windows)	49	Enabling communications between database partition servers	89
Installing DB2 online documentation (Windows)	50	Enabling the execution of remote commands (UNIX).	90
Installing Single-Partition Database Environments on the AIX Platform	52	Enabling Control Center administration (UNIX).	91
Installation overview for DB2 servers (UNIX).	52	Applying the latest FixPak	92
Installing DB2 servers on AIX	54	Verifying a partitioned database server installation (UNIX).	93
Installation requirements for DB2 servers (AIX)	55	Installing DB2 online documentation (UNIX).	94
Memory requirements for servers (UNIX)	56	Chapter 3. Installing DB2 Personal Edition	97
Disk requirements for DB2 servers (UNIX)	57	Installing DB2 Personal Edition (Windows)	97
Mounting the DB2 CD-ROM (AIX)	58	DB2 Personal Edition installation overview (Windows)	98
Starting the DB2 Setup wizard for a DB2 server installation (UNIX)	58	Installation requirements for DB2 Personal Edition (Windows)	100
Applying the latest FixPak	61	Memory requirements for DB2 Personal Edition (Windows)	102
Verifying the installation using the command line processor (CLP).	61	Disk requirements for DB2 Personal Edition (Windows)	102
Installing DB2 online documentation (UNIX).	62	Extending the directory schema (Windows 2000 and Windows .NET)	103
Installing Multipartition Database Environments on the AIX Platform	64	User accounts for installation and setup of DB2 Personal Edition	104
Installation overview for a partitioned DB2 server (UNIX)	64	Starting the DB2 Setup wizard (Windows)	105
Installing a partitioned DB2 server (AIX)	67	Applying the latest FixPak.	108
Installation requirements for partitioned DB2 servers (AIX)	69	Verifying the installation using the command line processor (CLP)	108
Memory requirements for partitioned DB2 servers (UNIX)	71	Installing DB2 online documentation (Windows)	109
Disk requirements for a partitioned DB2 server (UNIX)	72	Chapter 4. Performing a Response File Installation.	113
Updating AIX environment settings for a partitioned DB2 installation	73	Response file installation types	113
Verifying that NFS is running (AIX)	75	Response files	114
Creating a DB2 home file system for a partitioned database system (AIX)	76	Available sample response files	115
Creating required users for a partitioned DB2 server installation (AIX)	78	Response file keywords	116
Mounting the DB2 CD-ROM (AIX)	80	Response file keywords	121
Copying the contents of the DB2 product CD-ROM to your computer	80	DB2 Control Server response file keywords for Windows operating systems	122
Installing a database partition server on the primary computer using the DB2 Setup wizard (UNIX)	81	Response file generator	123
		db2rspgn - Response file generator	124

Killing DB2 processes during an interactive installation	125	Migrating the Satellite Control Server (UNIX)	155
Killing DB2 processes during a response file installation	126	Migration restrictions	156
Response file installation of DB2 on UNIX	126	Migration recommendations	158
Creating a response file on UNIX	128	Backing up databases before DB2 migration	160
Performing a response file installation on UNIX	129	Space considerations for DB2 migration	161
Response file installation of DB2 on Windows	129	Recording system configuration settings before DB2 migration	162
Making DB2 files available for a response file installation	130	Verifying that your databases are ready for migration	162
Setting up shared access to a directory on Windows	131	Taking a V6 or V7 DB2 server offline for DB2 migration	164
Creating a response file on Windows	132	Migrating instances (UNIX)	165
Running setup with the response file from the client workstation on Windows	133	Migrating the DB2 Administration Server (DAS)	167
Installing DB2 products using Microsoft Systems Management Server (SMS)	135	Migrating databases	167
Importing the DB2 install file into SMS	136	Migrating DB2 Personal Edition	169
Creating the SMS package on the SMS server	137	Migrating DB2 Personal Edition (Windows)	169
Distributing the DB2 installation package across your network	138	Preparing to migrate DB2 Personal Edition (Windows)	170
Configuring remote access to a server database	140	Migrating databases on DB2 Personal Edition (Windows)	173
Configuring db2cli.ini for a response file installation	142		
Exporting and importing a profile	142		
Chapter 5. Migrating a Satellite Environment	143	Chapter 6. Adding Existing DB2 Servers to a Satellite Environment	175
Migrating Servers on Windows	143	Adding Existing DB2 Servers to a Version 8 Satellite Environment	175
Migrating the Satellite Control Server (Windows)	143	Sample Scenario for Adding Existing DB2 Servers to a Satellite Environment	176
Migration restrictions	144	Adding Existing DB2 Servers to the Satellite Environment Using Your Own Scripts	178
Migration recommendations	146	Adding New and Existing DB2 Servers to the Satellite Environment Using a Fix Batch	179
Backing up databases before DB2 migration	147	Setting the Execution Starting Point to the Next Batch Step	182
Space considerations for DB2 migration	148		
Recording system configuration settings before DB2 migration	149		
Changing the diagnostic error level before DB2 migration	150		
Verifying that your databases are ready for migration	151		
Taking a V6 or V7 DB2 server offline for DB2 migration	152		
Migrating databases	153		
Migrating Servers on UNIX	155		
		Part 2. Administering a Satellite Environment	185
		Chapter 7. Batches and Application Versions	187
		Batches	187
		Batch Steps	188
		Components of a Batch Step	188
		Parameterized Scripts	193
		Batch Modes	194

Application Versions in a Satellite Environment	196	Requirement to Catalog Systems, Instances, and Databases on the Control Center	230
Group Batches	197	Cataloging the Satellite Control Server and the Satellite Control Database on the Control Center	231
Group Batches in the Test-Production Cycle Execution of Test Batch Steps by Test Satellites	200	Cataloging the Model Office on the Control Center	231
Promotion of Test Batch Steps to Production Batch Steps	201	Cataloging Remote Instances and Databases on the Model Office	232
Life Cycle of an Application Version	201	Cataloging Remote Instances and Databases on the Model Office Using a Customized Client Profile (Windows)	233
Example Test-Production Cycle of a Setup Batch	204	Cataloging Local Instances on the Model Office	234
Levels of an Application Version	206	Completing the Setup of the Model Office Cataloging Instances and Databases on Test Satellites	236
Test, Production, and Obsolete States of Application Version Levels	208	Setting up a Test Satellite Using a Client Profile	237
States of an Application Version	213	Cataloging Instances and Databases on Production Satellites	237
Update Batch in a Test Level of an Application Version	213		
Promoting a Test Level to a Production Level in an Application Version	214	Chapter 10. Setting up and Testing Your Satellite Environment	239
Creating a Test Level from a Production Level in an Application Version	215	Setting up and Testing a Satellite Environment	239
Obsoleting a Production Level in an Application Version	216	Preparing for a Test Synchronization	240
Relationships Between Batches and Batch Steps	217	Creating the User ID and Authentication Credential Required for Satellite Synchronization	242
Storage of Script on a Satellite During a Synchronization Session.	218	Granting Privileges on the Satellite Control Database	243
Fix Batches	219	Recommended Privileges to Grant on the Satellite Control Database	245
		Recommended Privileges to Grant on Stored Procedures and Bind Files	247
		Creating a Group for the Satellites	247
		Creating Test Satellites in the Satellite Administration Center	248
		Creating the Satellite Authentication File on a Satellite	249
		Setting the Application Version on a Satellite	249
		Setting the DB2SATELLITEID Registry Variable on a Satellite	250
		Verifying the Setup Before the Test Synchronization	252
		Testing the Synchronization Capability of a Satellite	253
		Creating and Testing Group Batches	254
		Creating Authentication Credentials.	255
Chapter 8. Authentication in the Satellite Environment	221		
Authentication Credentials.	221		
Authentication Credentials Stored at the Satellite Control Server	222		
Storage of Authentication Credentials on Satellites	223		
Creation and Maintenance of Authentication Credentials on a Satellite	223		
Authentication with Target Servers for Script Execution	224		
Password Change Management	224		
Managing Password Changes for Access to the Satellite Control Server.	225		
Managing Password Changes at Target DB2 Servers	226		
Chapter 9. Cataloging Instances and Databases	229		
Requirement to Catalog Instances and Databases in the Control Center Instance	229		

Creating Execution Targets	256	Binding Utilities on the Satellite to the Satellite Control Database	280
Creating an Application Version for a Group	257	Programming the Application	281
Creating Level 0 of an Application Version	258	Programming a Synchronizing Application	281
Editing Level 0 of an Application Version to Create or Modify Group Batches	259	Setting the Application Version with the db2SetSyncSession API	282
Changing Batch Steps in a Group Batch	260	Retrieving the Application Version with the db2GetSyncSession API	283
Testing Group Batches	262	Testing the Ability of the Satellite to Synchronize Using the db2SyncSatelliteTest API	284
Enabling Test Satellites to Execute Test-Level Batches	262	Managing Synchronization Sessions Using APIs	285
Synchronizing Test Satellites to Execute Test-Level Batches	263	Building and Running Synchronizing Applications	289
Checking the Results of the Synchronization Session	264	Building and Running Your Synchronizing Application	289
Fixing Problems Caused by Test-Level Group Batches	265	Using the DB2 Synchronizer Application	290
Promoting the Batches of Test Level 0 to Production	267		
Setting the Execution Starting Point for a Satellite	267		
Chapter 11. Working with the Model Office	269	Chapter 13. Recovering the Satellite Environment	293
The Model Office and the Development and Acceptance-Testing Phase	269	Recoverable Elements in a Satellite Environment	293
Role of the Model Office in the Development and Acceptance Testing Phase	269	Recovering Control Information	294
Characteristics of a Model Office	270	Recovery of Control Information	294
Installing and Setting up the Model Office	271	Recovering the Control Center Directories	295
Synchronizing the Model Office to Test the Group Batches	273	Recovering the Satellite Control Server and the Satellite Control Database	296
The Model Office and the Production-Deployment and Post-Deployment Phases	274	Recovering the Test Environment	298
Model Office During the Production-Deployment Phase	274	Recovery of Satellites in the Production Environment	306
Model Office in the Post-Deployment Phase	275		
Uses of the Model Office During the Post-Deployment Phase	275	Chapter 14. Performing a Mass Deployment	309
Chapter 12. Developing a Synchronizing Application	277	Performing a Mass Deployment	309
How a Synchronizing Application Works	277	How to Perform a Mass Installation	310
Setting up the Environment and the Development Machine	278	Performing a Mass Installation	310
Setting up the Environment for a Synchronizing Application	278	Role of the Model Office in a Mass Installation	312
Cataloging the Satellite Control Database on a Development Machine	279	Customizing the Generated Response File for a Mass Installation	313
		Preparing and Using Your Distribution Media for Mass Installation	314
		Customizing the Operating Environment of Each Satellite During Mass Installation .	316
		Completion of the Satellite Setup During a Mass Installation	317
		How to Perform a Mass Copy	317
		Performing a Mass Copy	318

DB2 Considerations for a Mass Copy	318
Application Data Considerations for a Mass Copy	319
Operating System Considerations for a Mass Copy	319
Completing the Mass Deployment	320
Installing a New Version of the Application on Group Satellites	320
Installing a New Version of the Application	320
Setting the New Application Version on the Satellite for the New Version of the Application	321
Creating and Testing the Group Batches for the New Version of the Application	322
Creating a Test System to Test the Deployment of the New Application	323
Deploying the New Application Version to the Production Satellites.	325
Monitoring Which Satellites Implemented the New Application Version	325
Chapter 15. Problem Determination	327
Installation Problems.	327
Location of Error Messages for Satellite Control Server Installation	328
Location of Error Messages for Satellite Installation	328
Configuration Problems That Prevent Synchronization	329
Synchronization Problems During Test Synchronizations	329
Synchronization Problems	332
Identifying and Fixing Failed Satellites.	336
Identifying and Fixing a Failed Satellite	337
Identifying the Failed Satellite	337
Obtaining Information About the Failure on the Satellite.	339
Assigning a Fix Batch to the Satellite	340
Debugging the Fix Batch	342
Returning the Repaired Satellite to Production	343
Running the DB2 Trace Facility on a Satellite	344
Satellite Software Version	345
Internal and External Error Return Codes for Batch Steps	345
Satellite Progress File	346
Recreating or Updating the satadmin.aut File on a Satellite	346

Determining Synchronization Errors when Log Details Are Truncated	347
--	-----

Part 3. Appendixes 349

Appendix A. Unique Characteristics of DB2 Satellite Edition Satellites.	351
Appendix B. Overview of the Satellite Control Tables	353
Appendix C. General Administration Tables	357
Appendix D. Naming rules	373
Appendix E. DB2 object naming rules	375
Appendix F. Workstation naming rules	377
Appendix G. User, userID and group naming rules	379
Appendix H. Changing the DB2 interface language (UNIX).	381
Appendix I. Changing the DB2 interface language (Windows)	383
Appendix J. DB2 Universal Database technical information	385
Overview of DB2 Universal Database technical information	385
Categories of DB2 technical information	385
Printing DB2 books from PDF files	393
Ordering printed DB2 books	394
Accessing online help	394
Finding topics by accessing the DB2 Information Center from a browser	396
Finding product information by accessing the DB2 Information Center from the administration tools	398
Viewing technical documentation online directly from the DB2 HTML Documentation CD.	399
Updating the HTML documentation installed on your machine	400
Copying files from the DB2 HTML Documentation CD to a Web Server.	402

Troubleshooting DB2 documentation search with Netscape 4.x.	402	DB2 Information Center for topics	407
Searching the DB2 documentation	403	Appendix K. Notices	409
Online DB2 troubleshooting information	404	Trademarks	412
Accessibility	405	Index	415
Keyboard Input and Navigation	405	Contacting IBM	425
Accessible Display	406	Product information	425
Alternative Alert Cues	406		
Compatibility with Assistive Technologies	406		
Accessible Documentation	406		
DB2 tutorials	406		

About This Book

This book provides the information necessary to:

- Install and migrate satellites.
- Set up and maintain a satellite environment

Who Should Use This Book

This book is intended primarily for administrators who implement and maintain a satellite environment. It can also be used by programmers who create applications that run on the satellites, as well as support personnel.

How This Book Is Structured

For general information about the satellite environment, see “Concepts” on page xiii. This section provides an overview of the satellite environment, and how the different components of this environment interact. The remainder of the book is divided into three parts. The first part contains information about how to install and migrate DB2 on the satellites:

- Chapter 1, “Installing the Satellite Control Server and Satellites” on page 3 describes how to install a satellite environment, and includes information on installing the satellite control server, satellites, and how to set up the satellite control server and satellite control database.
- Chapter 2, “Installing DB2 Workgroup Server Edition and DB2 Enterprise Server Edition” on page 11 describes how to install the DB2 Universal Database Workgroup Server Edition and Enterprise Server Edition products. In a satellite environment, DB2 Enterprise Server Edition on Windows-based platforms and AIX can be used as a satellite control server. DB2 Enterprise Server Edition and DB2 Workgroup Server Edition on Windows-based platforms can be used as satellites.
- Chapter 3, “Installing DB2 Personal Edition” on page 97 describes how to install the DB2 Universal Database Personal Edition product. In a satellite environment, DB2 Personal Edition on Windows-based platforms can be used as a satellite.
- Chapter 4, “Performing a Response File Installation” on page 113 describes how to install DB2 by using a response file.
- Chapter 5, “Migrating a Satellite Environment” on page 143 describes how to migrate existing DB2 systems to DB2 Universal Database Version 8.
- Chapter 6, “Adding Existing DB2 Servers to a Satellite Environment” on page 175 describes how to add existing DB2 servers into a satellite environment.

The second part contains information about the satellite environment:

- Chapter 7, “Batches and Application Versions” on page 187 describes how to create and modify the batches that set up and modify the configuration on the satellites.
- Chapter 8, “Authentication in the Satellite Environment” on page 221 describes how authentication works in the satellite environment.
- Chapter 9, “Cataloging Instances and Databases” on page 229 describes how to catalog the different DB2 targets against which the satellites must authenticate for synchronization.
- Chapter 10, “Setting up and Testing Your Satellite Environment” on page 239 describes how to set up and test the satellite environment.
- Chapter 11, “Working with the Model Office” on page 269 describes how you can use the model office during the different phases of the deployment, from the development phase through the post-deployment phase.
- Chapter 12, “Developing a Synchronizing Application” on page 277 describes using the application programming interfaces that are specific to the satellite environment.
- Chapter 13, “Recovering the Satellite Environment” on page 293 describes the recovery considerations that apply to the satellite environment.
- Chapter 14, “Performing a Mass Deployment” on page 309 describes how to perform a mass roll out of your production satellites.
- Chapter 15, “Problem Determination” on page 327 describes typical problems, and how to recover from them.

The third part contains the appendixes:

- Appendix A, “Unique Characteristics of DB2 Satellite Edition Satellites” on page 351 describes the differences between satellites running DB2 Satellite Edition and satellites that run other DB2 servers.
- Appendix B, “Overview of the Satellite Control Tables” on page 353 provides an overview of the tables of the satellite control database, while Appendix C, “General Administration Tables” on page 357 describes these tables in detail.
- Appendix D, “Naming rules” on page 373 describes the conventions for naming objects in a DB2 environment, while Appendix E, “DB2 object naming rules” on page 375, Appendix F, “Workstation naming rules” on page 377, and Appendix G, “User, userID and group naming rules” on page 379 provide more specific details.
- Appendix H, “Changing the DB2 interface language (UNIX)” on page 381 and Appendix I, “Changing the DB2 interface language (Windows)” on page 383 describe the code page and language support provided by DB2.
- Appendix J, “DB2 Universal Database technical information” on page 385 describes the DB2 library.

Concepts

Satellite Environment	xiii	Application Versions and Batches in a	
Satellite Control Server	xiv	Satellite Environment	xx
Satellite Control Database	xv	Satellite Synchronization	xxi
Groups in a Satellite Environment	xvi	Administration of Groups of Satellites	xxiii
Satellites.	xvii	Satellite Administration Center	xxiv
Model Office and Its Role in a Satellite		Example of a Satellite Environment	xxv
Environment	xviii		

Satellite Environment

The *satellite environment* is an environment in which you can administer large numbers of DB2[®] servers from a central control site.

In the satellite environment, you set up collections of DB2 servers. Each collection is known as a *group*. Each DB2 server that belongs to a group is called a *satellite*. You use groups to organize satellites that have shared characteristics into single entities. The characteristics that many satellites could share are the application that runs on them, and the database definition that supports the application.

Note: The *database definition* is the entire setup of DB2, including, but not limited to, the instance, the database manager configuration parameter values, the database design, and the database configuration parameter values.

In a group, the satellites are similar in terms of their database definition, usage, and purpose. For example, assume that there are two groups in your organization, Finance and Personnel. The Finance group would require one application and database definition, while the Personnel group would require a different application and database definition.

By grouping the satellites together, you administer small numbers of groups, which may contain hundreds, if not thousands of satellites, rather than having to administer each satellite individually. If you acquire additional DB2 servers to perform the same function as the satellites of an existing group, you can add them to that group. The administration solution provided by the satellite environment is fully scalable.

In the satellite environment, the setup and maintenance of any database definition is accomplished by sets of scripts known as *batches*. Because the database definition can be different for each group, each group has its own set of batches. The batches that are specific to a group are known as *group batches*.

Within a group, satellites may run different versions of the application, and each version of the application may require its own database definition and data. Group batches are always associated with a particular *application version*. The application version represents the database definition that supports a particular version of the end-user application.

Because a satellite belongs to a group, it will have the same application as the other satellites in the group. Depending on its version of the application, however, it may share its database definition with only a subset of the satellites that belong to its group.

Information about the satellite environment is stored in a central database known as the *satellite control database*. This database records, among other things, which satellites are in the environment, the group each satellite belongs to, and which version of the application a satellite is running. It also records the group batches that the satellites execute. This database is on a DB2 server that is known as the *satellite control server*.

To set up and maintain its database definition, each satellite connects to the satellite control database to download the batches that correspond to its version of the application. The satellite executes these batches locally, then reports the results back to the satellite control database. This process of downloading batches, executing them, then reporting the results of the batch execution is known as *synchronization*. A satellite synchronizes to maintain its consistency with the other satellites that belong to its group and that run the same version of the application.

Related concepts:

- “Satellite Control Server” on page xiv
- “Satellite Control Database” on page xv
- “Groups in a Satellite Environment” on page xvi
- “Satellites” on page xvii
- “Application Versions and Batches in a Satellite Environment” on page xx
- “Satellite Synchronization” on page xxi
- “Satellite Administration Center” on page xxiv

Satellite Control Server

The satellite control server maintains the satellite control database, which records information about the satellite environment, and manages the synchronization process for satellites. You can use any supported DB2[®] server on any of the supported operating systems as a satellite control server. This

means that you can use tools such as the Control Center to administer it. You can also use the audit and trace facilities on this DB2 server.

Note: Currently, only servers running DB2 Universal Database Enterprise Server Edition on Windows® and AIX® platforms can function as a satellite control server.

On Windows-based platforms, the satellite control server is created automatically when you select **Satellite administration capability** during a typical install, or when you select the Satellite Control Server component as part of a custom installation.

On AIX, you create the satellite control server after installing the Satellite Control Server component, by using the normal procedure for creating an instance. To install the Satellite Control Server component, you must select the **Satellite administration capability** check box during a typical installation. If you are performing a custom installation, ensure that the Satellite Control Server component is selected.

When the satellite control server is created on Windows-based platforms, the satellite control database, SATCTLDB, is also automatically created. On AIX, you create this database after creating the satellite control server instance.

Related concepts:

- “Satellite Control Database” on page xv

Related reference:

- “Software Requirements for the Satellite Control Server” on page 5

Satellite Control Database

The information that is required by the administrator and the satellite control server to configure and maintain satellites is stored in tables in the satellite control database. A relational database system is best suited to manage and maintain the integrity of the control data that is required to manage large numbers of satellites because:

- You can maintain (back up, for example) the data by using standard tools.
- The data is easily accessible by authorized users.
- You can use load balancing and fault tolerance through replication. In addition, you can also use failover products to ensure availability.

The satellite control database always has the same name, SATCTLDB. The fixed name simplifies detection of the database by the Control Center, and the Satellite Administration Center. In addition, it also simplifies the synchronization process because the satellite control database is easier to find.

The tables in the satellite control database are used and maintained in two ways:

- The Satellite Administration Center, the graphical interface that you can use to update and query the information in the tables.
- Stored procedures that the synchronizing application invokes when the satellite synchronizes with the satellite control server. The stored procedures both read and update information in the tables.

The Satellite Administration Center is the supported tool for accessing and updating the control information in the SATCTLDB database. If, however, it is absolutely necessary, you can directly manipulate the tables by using SQL. You may want to do this, for example, if you are deploying a large group of satellites. In this situation, it may prove faster to INSERT into the tables, rather than using the Satellite Administration Center to create hundreds, if not thousands, of satellites.

Notes:

1. Because of the referential constraints and triggers that protect the integrity of the data in the satellite control database, it is typically difficult to update the table data directly.
2. Starting with Version 8, the satellite control database can be a partitioned database.

Related concepts:

- “Satellite Control Server” on page xiv
- “Satellite Administration Center” on page xxiv
- Appendix B, “Overview of the Satellite Control Tables” on page 353

Groups in a Satellite Environment

Every satellite in the satellite environment must belong to a group (and cannot belong to more than one group). A group is a collection of satellites that share one or more characteristics. One shared characteristic is the business function of the satellites. For example, you may have many similar DB2[®] servers that function as departmental servers. Based on this similarity, these DB2 servers may be candidates for membership in the same group.

A satellite can only be associated with a single application version at a time. However, within a group, not all satellites need to be associated with the

same application version. This characteristic allows you to stage the deployment of the application. The different versions of the application are supported by the group batches that are associated with different application versions.

Note: The term *group*, as used in the satellite environment, is *not* associated with operating system or security groups.

Related concepts:

- “Satellites” on page xvii
- “Model Office and Its Role in a Satellite Environment” on page xviii
- “Application Versions and Batches in a Satellite Environment” on page xx
- “Administration of Groups of Satellites” on page xxiii

Satellites

A satellite is any DB2[®] server that is both a member of a group and synchronizes with a satellite control server to maintain its database definition and data.

Along with DB2, the satellite will run the business application that your end users require. The hardware on which DB2 and your application run can be any computer on which any of the supported operating systems is running.

Notes:

1. Servers running DB2 Universal Database Personal Edition, DB2 Universal Database Workgroup Server Edition, and DB2 Universal Database Enterprise Server Edition on supported Windows-based platforms can be used as satellites.
2. If you are using a Version 8 satellite control server, all satellites that synchronize with it must running DB2 Universal Database Version 8.

There are two types of satellites in the satellite environment: *test satellites* and *production satellites*. You use test satellites to test the group batches that set up and maintain the database definition that supports the application. When the group batches are fully tested, and produce satisfactory results, you promote them for use by the production satellites of the group.

Typically, you assign test satellites to your development environment. That is, the test satellites will not manage any active data that is required for business operations.

You assign production satellites to the users that you support. Unlike test satellites, production satellites are not used for testing purposes. Instead, these

satellites execute the tested group batches that have been put into production, which results in them having a stable database definition to support the application. Because production satellites have a stable database definition, they manage the active data that is required for business operations.

If you determine that the database definition on the production satellites is no longer adequate to support the requirements of the application, you modify the database definition on the test satellites to address the problem that is being reported. When you are satisfied with the results of the modification in your development environment, you make the modification available to the production satellites.

When you create a satellite, it is, by default, a production satellite. You must decide which satellites that you want to use as test satellites. This allows you to exercise full control over testing. For information about how to define a satellite as a test satellite, refer to the online help that is available from the Satellite Administration Center.

Note: Because test satellites are not used to maintain active data, you should not change them to production satellites during the lifetime of the application.

Related concepts:

- “Satellite Control Server” on page xiv
- “Groups in a Satellite Environment” on page xvi
- “Model Office and Its Role in a Satellite Environment” on page xviii
- “Satellite Synchronization” on page xxi

Related tasks:

- “Configuring a satellite as a test satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Model Office and Its Role in a Satellite Environment

A model office is a special member of the test satellites in the group. Typically, you will have one model office for each version of the application that you have deployed in the group. You can use a model office for a variety of purposes:

- To model your initial deployment of the group.

When the model office is set up, you can create a response file that is based on the model office. You can use the response file to install large numbers of satellites.

- To test the changes that are required to the database definition that supports a version of the application that is already in production.
The model office provides a representative database, which you can work with using tools such as the Control Center and the Satellite Administration Center. Using the model office, you can generate scripts from the different wizards, notebooks, and windows available in the Control Center. You verify the behavior of the scripts by submitting them to execute against the model office. This means that the production environment is almost entirely isolated from the changes that you make to the model office. When you are satisfied with the database definition changes that the scripts produce on the model office, you can promote the scripts to production. The changes will then be executed by the group’s production satellites.
- To provide a representation of a typical satellite in the group.
Because the model office represents a typical satellite, it can be useful for problem determination. If a problem occurs on a satellite, you can use the model office, or a copy of it, to reproduce the problem and determine how to fix it.
- To deploy a new version of your application. You can use the model office to verify that the installation of the new version of the application provides the correct results, and that the batches produce the expected database definition for the application.

The following figure provides an overview of how the Control Center, the satellite control server, and the model office can interact.

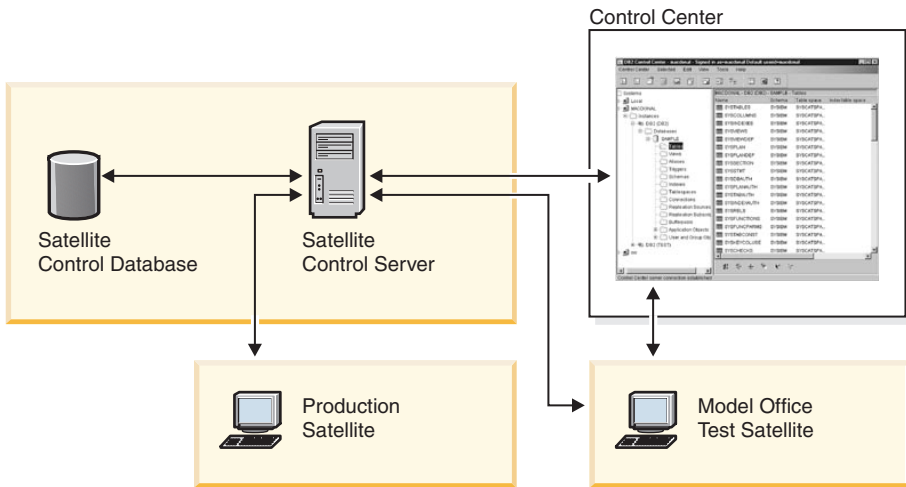


Figure 1. Model Office

Related concepts:

- “Groups in a Satellite Environment” on page xvi

- “Satellites” on page xvii
- “Role of the Model Office in the Development and Acceptance Testing Phase” on page 269
- “Characteristics of a Model Office” on page 270

Related tasks:

- “Installing and Setting up the Model Office” on page 271

Application Versions and Batches in a Satellite Environment

Although the satellites of a group run the same application, they do not necessarily have to run the same version of this application. Each version of the application may require a database definition that is different from that required by other versions. To set up and maintain the database definition and the data to support a particular version of the application, you use group batches for the application version. Each application version of a group is associated with its own batches.

Each batch is a collection of one or more batch steps. You create these batch steps to set up and maintain the database definition for the application version. The batch step is executed on the satellite when the satellite synchronizes.

A batch step is made of the following components:

- A *script*. The script can contain one or more DB2® commands, SQL statements, or operating system commands (though the script cannot contain .
- An *execution target*. The script that you create can execute against a DB2 instance, DB2 database, or on the operating system on the satellite. The DB2 instance, DB2 database, or operating system against which the script executes is known as an execution target.
- An *authentication credential*. Before a script can execute against a DB2 instance or DB2 database, it must be authenticated. That is, the script requires the combination of a user ID and password so that the satellite can attach to the instance or connect to the database. This combination of user ID and password is known as an authentication credential.
- A *success code set*. The execution of a script is considered to be successful if its return code is within a set of return codes that you predefine for that script. This set of codes is known as a success code set.

The batch steps within a batch are always executed in the sequence in which they appear in the batch. When one batch step within a batch is executed successfully, the next batch step is executed. If, as defined by the success code

set, an error occurs when a satellite is executing a batch step, that satellite stops executing its group batches, and reports an error back to the satellite control server. When the error is fixed, the satellite can continue executing from the batch step that caused the error.

The application version is set at the satellite, typically during the installation and configuration of the application on the satellite. When a satellite synchronizes, it reports its application version to its satellite control server before it downloads and executes the scripts associated with the group batches for that application version.

Related concepts:

- “Batches” on page 187
- “Batch Steps” on page 188
- “Batch Modes” on page 194
- “Application Versions in a Satellite Environment” on page 196
- “Life Cycle of an Application Version” on page 201
- “Components of a Batch Step” on page 188
- “Authentication Credentials” on page 221
- “Creation and Maintenance of Authentication Credentials on a Satellite” on page 223

Related tasks:

- “Creating and Testing Group Batches” on page 254
- “Creating Authentication Credentials” on page 255
- “Creating Execution Targets” on page 256
- “Creating an Application Version for a Group” on page 257

Satellite Synchronization

The satellites of a particular group need to be in a state consistent with the other satellites of its group. The consistent state can be accomplished with synchronization.

The following figure provides a high-level view of how a satellite synchronizes. Before the satellite can synchronize, it must connect to the network on which the satellite control server resides.

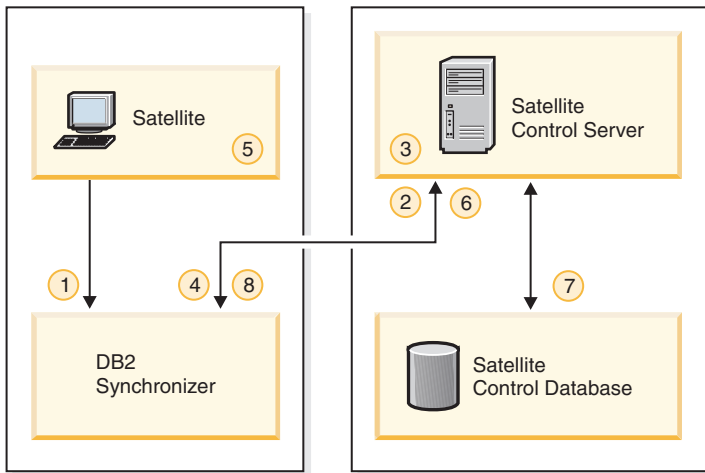


Figure 2. Satellite Synchronization

The satellite synchronizes as follows:

1. The synchronization function is invoked on the satellite. The invocation can be from your application (if it calls the `db2SyncSatellite` API), or from the DB2[®] Synchronizer application that is provided with DB2.

When the synchronizer function is invoked, steps 2 through 8 occur automatically. No manual intervention is required.

The satellite is only connected to the satellite control database for step 3, and for step 7.
2. The satellite connects to the satellite control database, where it is authenticated.
3. After authentication occurs, the satellite control server checks which group the satellite belongs to, and the version of the application that the satellite is executing. The satellite control server uses this information to determine which batches the satellite should execute, and which batch steps should be executed, if any. At this time, other events may also occur:
 - a. If the satellite did not upload the results of its previous synchronization session to the satellite control database, the satellite writes the results to the satellite control database.
 - b. If any of the scripts in a batch to be downloaded are parameterized, the satellite control server instantiates the script with the values that are appropriate for the satellite.
 - c. When steps 3a and 3b are complete (if required), the satellite control server retrieves the scripts that the satellite is to execute, and the satellite downloads them. When this occurs, the tables in the satellite control database are updated to indicate that the satellite has obtained the batches that apply to it.

4. The synchronizer function drops the connection with the satellite control database.
5. The satellite executes the batches that it downloaded.
6. After executing the batches, the synchronizer function again connects to the satellite control database.
7. The synchronizer function updates the log information in the satellite control database with the results of the execution of the different steps of the batches. The log information provides details about the execution of the batch steps. For information about these logs, refer to the online help that is available from the Satellite Administration Center.
8. The synchronizer function drops the connection with the satellite control database.

Note: The number of satellites that can concurrently synchronize with the satellite control server is determined by the value of the *maxappls* database configuration parameter at the satellite control database.

Related concepts:

- “Parameterized Scripts” on page 193
- “Storage of Script on a Satellite During a Synchronization Session” on page 218
- “Satellite Progress File” on page 346

Related tasks:

- “Viewing logs for a group : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Viewing details of a log entry : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Viewing log details : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Related reference:

- “Maximum Number of Active Applications configuration parameter - maxappls” in the *Administration Guide: Performance*
- “db2sync - Start DB2 Synchronizer Command” in the *Command Reference*
- “db2SyncSatellite - Sync Satellite” in the *Administrative API Reference*

Administration of Groups of Satellites

Within a group, satellites can run the same application, have a similar database definition, and perhaps the same execution environment.

The group can also contain satellites that run a different version of the application. Because the database definition that supports each version of the application is set up and maintained by the batches of a specific application version, you can deploy different versions of the application. This enables you to stage the deployment of a new version of the application within the group.

Because satellites are organized by group, you administer at the group level, and not at the individual satellite level. This greatly simplifies administration. Instead of having to manage hundreds, if not thousands, of satellites separately, you manage the group to which they belong. The group batches that maintain the database definition for a particular version of an application are associated with the application version. These group batches are organized into application versions for each version of the application running on the satellites of the group.

When you create new satellites for the satellite environment, you add them to the group that is already running the application that the new satellites will run. When these satellites synchronize for the first time, they will download and execute the group batches that apply to the version of the application that they are running. You do not have to perform any special tasks to integrate these satellites into the environment. This means that the administration model that you use to set up and maintain the satellite environment is fully scalable. The groups that you set up can contain as many satellites as your business requires.

Related concepts:

- “Groups in a Satellite Environment” on page xvi
- “Satellites” on page xvii
- “Application Versions and Batches in a Satellite Environment” on page xx

Satellite Administration Center

The Satellite Administration Center is a collection of graphical tools that is available from the Control Center. You use the Satellite Administration Center to set up and maintain satellites, groups, and the batches that the satellites execute when they synchronize.

Before you can use the Satellite Administration Center, ensure that you add the satellite control server and the satellite control database to the Control Center object tree.

You can use the Satellite Administration Center when the Control Center detects that any of the instances available on it contain a satellite control database. You can open the Satellite Administration Center from:

- The pop-up menu associated with the instance that contains the satellite control database
- The pop-up menu associated with the satellite control database
- The Control Center tool bar
- The Control Center Tools menu

For information about using the Satellite Administration Center, refer to the online help provided with it.

Note: You should only administer the satellite environment from the Satellite Administration Center; it is recommended that you do not try to administer the satellite environment by directly modifying the tables in the satellite control database.

Related concepts:

- “Satellite Control Database” on page xv

Related tasks:

- “Cataloging the Satellite Control Server and the Satellite Control Database on the Control Center” on page 231
- “ : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Open the Satellite Administration Center : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Example of a Satellite Environment

The following figure shows a possible setup of the satellite environment. In the example, the development environment, which includes the Control Center, the satellite control server and satellite control database, as well as the model office and test satellites, is almost entirely separate from the production environment. You use the development environment to both create and test the batches that you want the production satellites to execute.

Development Environment

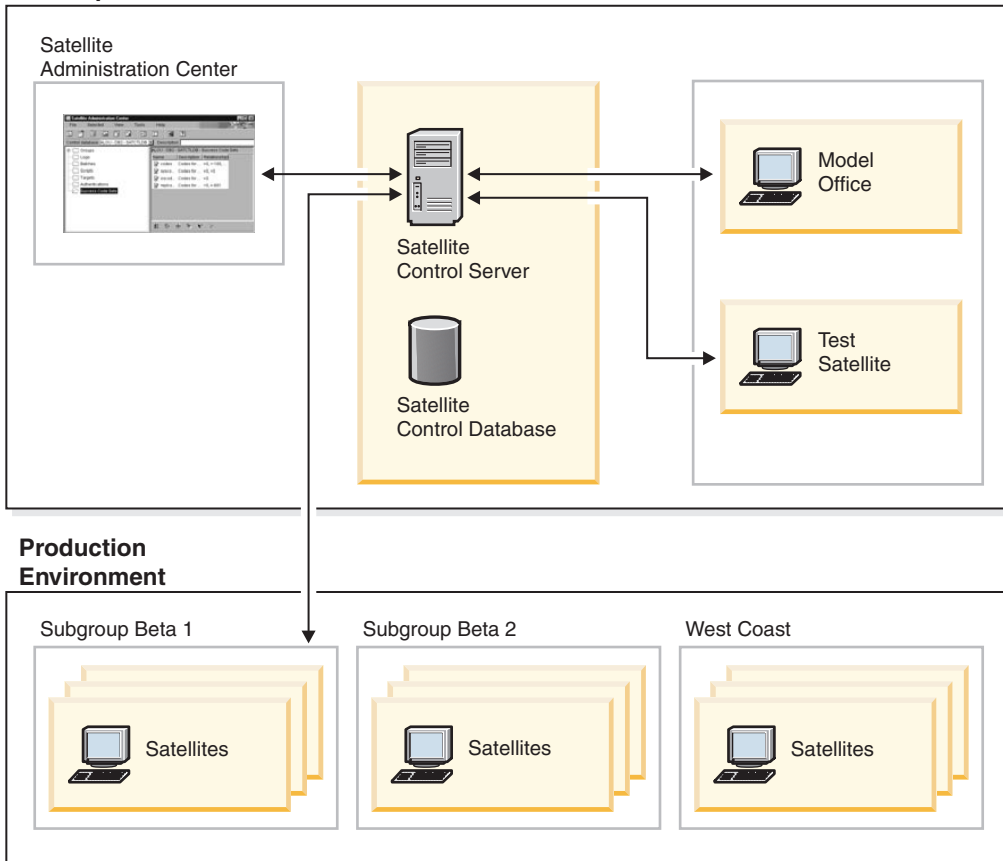


Figure 3. Satellite Environment

In the previous figure, all the production satellites in the production environment belong to the same group, but belong to different subgroups. You can specify that a satellite belongs to a specific subgroup when you either create or edit the satellite with the Satellite Administration Center. You can use subgroups to stage the deployment of the first version of the application.

When rolling out the first version of the application, you stage the deployment to control which satellites can synchronize (that is, which satellites can execute the group batches). You also stage the deployment to test whether the database definition is appropriate for the application in the production environment. While the group batches may produce correct results on the model office and test satellites of the development environment, the active data of the production environment may indicate that the group batches have to be modified. For example, in the previous figure, the subgroup Beta 1 is the first stage of the deployment, that is, only the Beta 1

subgroup can synchronize with the satellite control server. Assume that you receive reports from the Beta 1 subgroup that the performance of the application is not satisfactory. You can address the application-performance problem, then, when the problem is resolved for the Beta 1 subgroup, continue by rolling out the Beta 2 subgroup. Because the Beta 1 and Beta 2 subgroups are running the same version of the application, they execute the same group batches of the same application version. This means that the Beta 2 subgroup is not likely to report the same problem as the Beta 1 subgroup. To stage the deployment of the first version of the application by subgroups, you enable the satellites, subgroup by subgroup, to execute the group batches.

You can also use subgroups to stage the deployment of the next version of the application. For example, assume that the Beta 2 and West Coast subgroups are running the first version of the application, and that you have tested the second version of the application on a model office or test satellite, then installed the new version of the application on the Beta 1 subgroup. In this situation, all the subgroups will be enabled to synchronize, and all will be maintaining active data. The difference is that when the Beta 1 subgroup synchronizes, it executes the group batches associated with the second application version, while Beta 2 and West Coast execute the group batches of the first application version. In this situation, you can use the Beta 1 subgroup both to determine whether the new version of the application is appropriate for your business requirements in the production environment, and whether the group batches that the Beta 1 subgroup executes produce satisfactory results.

Related concepts:

- “Satellite Control Server” on page xiv
- “Satellite Control Database” on page xv
- “Groups in a Satellite Environment” on page xvi
- “Satellites” on page xvii
- “Model Office and Its Role in a Satellite Environment” on page xviii
- “Application Versions and Batches in a Satellite Environment” on page xx
- “Satellite Synchronization” on page xxi
- “Administration of Groups of Satellites” on page xxiii
- “Satellite Administration Center” on page xxiv

Related tasks:

- “Installing and Setting up the Model Office” on page 271
- “Performing a Mass Deployment” on page 309

Part 1. Installing and Migrating a Satellite Environment

Chapter 1. Installing the Satellite Control Server and Satellites

Preparation for the Installation of a Satellite Environment	3	Satellite Control Database Considerations	6
Installing and Setting Up the Satellite Control Server and Satellite Control Database	5	Setting Up the Satellite Control Server on AIX	7
Disk Requirements for the Satellite Control Server	5	Creating the DB2CTLSV Instance on AIX.	7
Software Requirements for the Satellite Control Server	5	Creating the SATCTLDB Database on AIX	8
		Customizing the SATCTLDB Database on Windows	9

The section that follows provides general information on installing a satellite environment.

Preparation for the Installation of a Satellite Environment

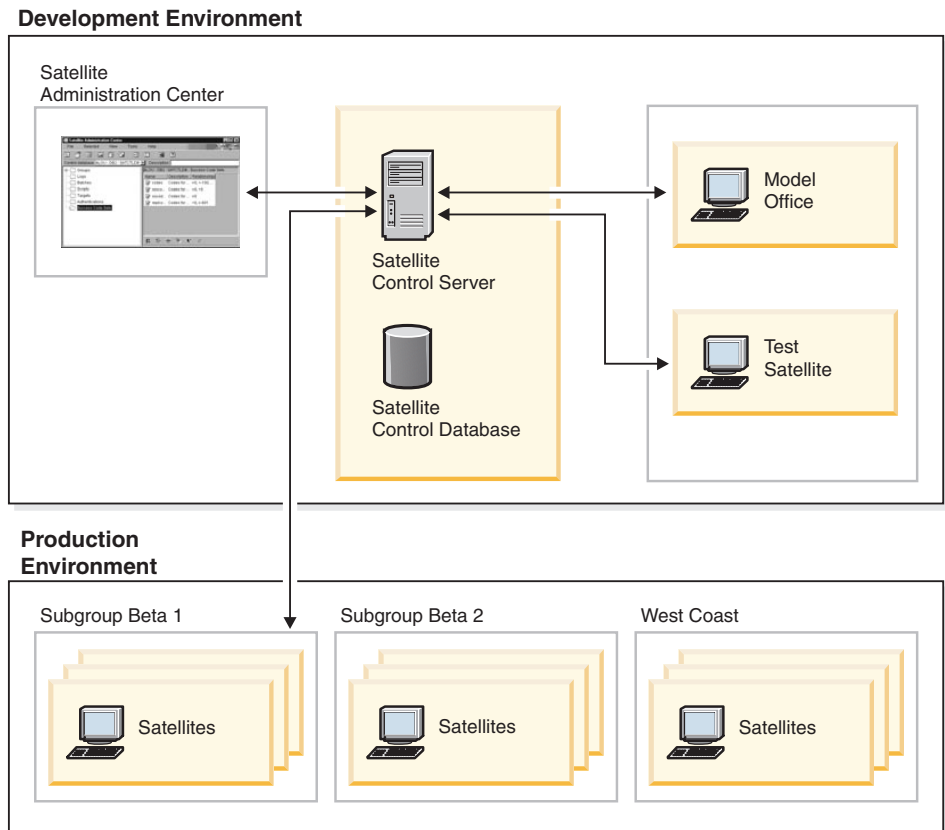
For a satellite environment, the installation scenario will vary from one organization to another. At a minimum, you will need to install the satellite control server, and the satellites:

- To install the satellite control server, you must install DB2[®] UDB Enterprise Server Edition with the Satellite Control Server component. You can install DB2 UDB Enterprise Server Edition on any supported Windows-based platform, or on AIX. In addition, the database environment that you install can be either a single-partition environment, or a multipartition environment (that is, the SATCTLDB database can be a partitioned database).
- To install the satellites, you can install DB2 UDB Personal Edition, DB2 UDB Workgroup Server Edition, or DB2 UDB Enterprise Server Edition on any supported Windows-based platform. When you install DB2, you must also install the Satellite Synchronization component. The Satellite Synchronization component is required for the satellite to synchronize with its satellite control server.

You can also install the Control Center, which includes the Satellite Administration Center, on workstations used for satellite environment administration.

Note: The Version 8 Satellite Administration Center only supports satellites that are running DB2 UDB Version 8.

The following diagram provides a schematic of the systems in your satellite environment:



Notes:

1. The Control Center, which includes the Satellite Administration Center, is assumed to be installed on workstations other than the satellite control server. At a minimum, you will need to install the DB2 Administration Client on these workstations, although any other DB2 product that includes the Control Center can also be installed.
2. If you want to administer the satellite environment from the same workstation on which you have installed the satellite control server, select the Control Center component in addition to the Satellite Control Server component when you install DB2 UDB Enterprise Server Edition.

Related concepts:

- “DB2 Enterprise Server Edition” in the *Quick Beginnings for DB2 Servers*
- “DB2 Workgroup Server Edition” in the *Quick Beginnings for DB2 Servers*
- “DB2 Personal Edition” in the *Quick Beginnings for DB2 Personal Edition*

Installing and Setting Up the Satellite Control Server and Satellite Control Database

The sections that follow provide information about how to install the satellite control server and the satellite control database

Disk Requirements for the Satellite Control Server

This topic describes the minimum amount of disk space that is required to install the Satellite Control Server component, create the DB2CTLSV instance, and create an empty satellite control database (SATCTLDDB). These estimates are in addition to the disk requirements for installing DB2 Enterprise Server Edition. These estimates do not include the disk space required for the operating system, application, or communication products. Consult each product's documentation for these values.

Windows-based platform

The recommended minimum disk space for the satellite control server, plus an empty satellite control database, is 13 MB.

AIX The recommended minimum disk space for the satellite control server, plus an empty satellite control database, is as follows:

- Under the *INSTHOME*/db2ctlsv and *INSTHOME*/sqllib directory, is 23.2 MB, where *INSTHOME* represents the home directory of the instance.
- Under the /usr directory, is 0.4 MB

As you add groups, satellites, and scripts, and use the satellite control database, the satellite environment will consume more disk space.

Note: Synchronization logs can consume rather large amounts of disk space unless they are periodically purged.

Related reference:

- “Disk requirements for a partitioned DB2 server (UNIX)” on page 72
- “Installation requirements for DB2 servers (Windows)” on page 17
- “Installation requirements for DB2 servers (AIX)” on page 55
- “Disk requirements for DB2 servers (UNIX)” on page 57
- “Disk requirements for DB2 servers (Windows)” on page 19
- “Disk requirements for a partitioned DB2 server (Windows)” on page 36

Software Requirements for the Satellite Control Server

This topic describes the software requirements for the installation of the satellite control server. The satellite control server requires that you must

configure the satellite control server instance, DB2CTLSV, to support TCP/IP, because it is the only communications protocol with which synchronization can be used.

The Satellite Administration Center, a component of the Control Center, will connect to the satellite control server's satellite control database to set up and maintain satellites, groups, and the batches that the satellites execute when they synchronize. If the Control Center, and hence the Satellite Administration Center, is running on a remote system, these connections can be made using NetBIOS, TCP/IP, Name Pipes, or APPC. At a minimum, the appropriate TCP/IP communications software must be installed.

Related reference:

- "Installation requirements for DB2 servers (Windows)" on page 17
- "Installation requirements for DB2 servers (AIX)" on page 55

Satellite Control Database Considerations

When the default satellite control database (SATCTLDB) is created, all its tables and indexes are created in the default tablespace, userspace1. For large deployments, where the satellite control server will be managing thousands of satellites, you can change this design to allow more control over the SATCTLDB database's disk usage and performance. Disk tuning considerations for the SATCTLDB database are similar to those for any application database.

The file `satctl.db.ddl` in the `sql1lib\misc` directory contains the DDL to define the SATCTLDB database. A copy of this file can be made and then modified to define the location and size of additional table spaces that will be used for the SATCTLDB database and to assign the SATCTLDB database's tables to these table spaces.

Note: When modifying this file, the table and index definitions should only be modified to add table space attributes. Do not make changes to the DDL, such as referential integrity definitions or trigger definitions, as the operation of the SATCTLDB database, the Satellite Administration Center and satellite synchronization may be adversely affected.

The installation process on Windows-based platforms creates the SATCTLDB database when the Satellite Control Server component is selected. The installation process on AIX® does not automatically create the SATCTLDB database. If you want to use additional table spaces for your SATCTLDB database, you should create a modified version of the `satctl.db.ddl` file, and then use it to create the SATCTLDB database that will support your environment.

Related concepts:

- “Disk-storage performance factors” in the *Administration Guide: Performance*
- “Satellite Control Database” on page xv

Setting Up the Satellite Control Server on AIX

Prerequisites:

The SATCTLDB database is not automatically created during the satellite control server installation on AIX. You should create a modified version of the `satctl.db.ddl` file, and use it to create the SATCTLDB database that will support your environment. The file `satctl.db.ddl` can be found in the `sql1lib/misc` directory.

Procedure:

To set up the satellite control server:

1. Create the DB2CTLSV instance.
2. Run the `satctl.db.ddl` file to create the SATCTLDB database.

Related concepts:

- “Satellite Control Server” on page xiv

Related tasks:

- “Creating the DB2CTLSV Instance on AIX” on page 7

Creating the DB2CTLSV Instance on AIX

This topic describes how to create the DB2CTLSV Instance on AIX.

Procedure:

To create the DB2CTLSV instance:

1. Log on as user with root authority.
2. Change to the directory where the DB2 CD-ROM is mounted by entering the following command:

```
cd /cdrom
```

where `/cdrom` represents the mount point of the CD-ROM drive.

3. Change to the `/cdrom/db2/aix` directory where the install image for the DB2 product that you want to install is located.
4. Enter the `./db2setup` command to start the DB2 installation program. The **DB2 Installer** window opens.

5. Use the **Tab** key to select the **Create** option. The **Create DB2 Services** window opens.
6. Select **Create a DB2 Instance**. The **DB2 Instance** window opens.
7. Rename the user name field with **db2ctlsv**. Use the default User ID and password. Change the home directory to `/home/db2ctlsv` to match the instance. Select **OK**. The **Fenced User** windows opens.
8. Select **OK** to accept the defaults. The **Create DB2 Services** window opens.
9. Select **OK**. The **Summary Report** window opens.
10. Select **Continue**.

Related tasks:

- “Setting Up the Satellite Control Server on AIX” on page 7
- “Creating the SATCTLDB Database on AIX” on page 8

Creating the SATCTLDB Database on AIX

This topic describes how to create the SATCTLDB database on AIX.

Procedure:

To create the SATCTLDB database, perform the following steps:

1. Log in as `db2ctlsv`.
2. Ensure that the database server has been started; the **db2start** command has been issued.
3. If you do not want the default SATCTLDB database created, copy and edit the `satctl.db.ddl` file to satisfy your requirements.
4. Enter the following command, from the `sqllib/misc` directory:

```
db2 -tf prdctl.db.ddl -z $HOME/prdctl.db.log
```

where `prdctl.db.ddl` represents the modified version of the DDL file and is located in the `sqllib/misc` directory.

5. Check the `prdctl.db.log` file for errors that may have been encountered during the creation of the SATCTLDB database.

Related tasks:

- “Setting Up the Satellite Control Server on AIX” on page 7
- “Creating the DB2CTLSV Instance on AIX” on page 7
- “Customizing the SATCTLDB Database on Windows” on page 9

Customizing the SATCTLDB Database on Windows

This topic describes how to customize the SATCTLDB database on Windows.

Prerequisites:

For large deployments where thousands of DB2 satellites will be managed by the satellite control server, you should use your own design, rather than the default, to allow more control over the SATCTLDB database's disk usage and performance. If you want to use your own design, you must first drop the default database and re-create the SATCTLDB database based on your customized DDL file.

Procedure:

To customize your design, perform the following steps:

1. Make a copy of the `satctl.db.ddl` file and customize it to meet your needs. For the purposes of this example, `prdctl.db.ddl` is used as the filename for the customized DDL file.
2. Open a command window by clicking on **Start**, then selecting **Programs** → **IBM DB2** → **Command Line Tools** → **Command Window**.
3. In the Command Window, ensure that the `DB2INSTANCE` environment variable is set to `DB2CTLSV`. To check the `DB2INSTANCE` environment variable enter the `SET DB2INSTANCE` command. The value returned must be `DB2INSTANCE=DB2CTLSV`. If the environment variable is not set to `DB2CTLSV`, enter the `SET DB2INSTANCE=DB2CTLSV` command to change its setting.
4. Drop the default SATCTLDB database that was created during the installation by entering the following command:

```
DROP DATABASE SATCTLDB
```
5. Change to the directory where you have stored your customized `prdctl.db.ddl` file.
6. Enter the following command:

```
db2 -tf prdctl.db.ddl -z prdctl.db.log
```

where `prdctl.db.ddl` represents your customized DDL file.

Check the `prdctl.db.log` file for errors that may have been encountered during the creation of the SATCTLDB database.

Related tasks:

- “Setting Up the Satellite Control Server on AIX” on page 7
- “Creating the DB2CTLSV Instance on AIX” on page 7

- “Creating the SATCTLDB Database on AIX” on page 8

Chapter 2. Installing DB2 Workgroup Server Edition and DB2 Enterprise Server Edition

Installing Single-Partition Database Environments on Windows-Based Platforms	12	Installing database partition servers on participating computers (Windows)	45
Installation overview for DB2 servers (Windows)	12	Applying the latest FixPak	48
Installing a DB2 server on Windows	14	Verifying a partition database server installation (Windows)	49
Installation overview for DB2 servers (Windows)	15	Installing DB2 online documentation (Windows)	50
Installation requirements for DB2 servers (Windows)	17	Installing Single-Partition Database Environments on the AIX Platform	52
Memory requirements for DB2 servers (Windows)	19	Installation overview for DB2 servers (UNIX).	52
Disk requirements for DB2 servers (Windows)	19	Installing DB2 servers on AIX	54
Extending the directory schema (Windows 2000 and Windows .NET)	20	Installation requirements for DB2 servers (AIX)	55
User accounts required for installation of DB2 servers (Windows)	21	Memory requirements for servers (UNIX)	56
Starting the DB2 Setup wizard for a DB2 server installation (Windows)	23	Disk requirements for DB2 servers (UNIX)	57
Applying the latest FixPak	25	Mounting the DB2 CD-ROM (AIX)	58
Verifying the installation using the command line processor (CLP).	26	Starting the DB2 Setup wizard for a DB2 server installation (UNIX)	58
Installing DB2 online documentation (Windows)	27	Applying the latest FixPak	61
Installing Partitioned Database Environments on Windows-Based Platforms	29	Verifying the installation using the command line processor (CLP).	61
Installing a partitioned DB2 server (Windows)	29	Installing DB2 online documentation (UNIX).	62
Installation overview for partitioned DB2 servers (Windows)	30	Installing Multipartition Database Environments on the AIX Platform	64
Installation requirements for a partitioned DB2 server (Windows)	33	Installation overview for a partitioned DB2 server (UNIX)	64
Memory requirements for a partitioned DB2 server (Windows)	35	Installing a partitioned DB2 server (AIX)	67
Disk requirements for a partitioned DB2 server (Windows)	36	Installation requirements for partitioned DB2 servers (AIX)	69
Preparing the environment for a partitioned DB2 server (Windows)	37	Memory requirements for partitioned DB2 servers (UNIX)	71
Extending the directory schema (Windows 2000 and Windows .NET)	39	Disk requirements for a partitioned DB2 server (UNIX)	72
Installing the instance owning database partition server (Windows)	40	Updating AIX environment settings for a partitioned DB2 installation	73
Verifying port range availability on participating computers	44	Verifying that NFS is running (AIX)	75
		Creating a DB2 home file system for a partitioned database system (AIX)	76
		Creating required users for a partitioned DB2 server installation (AIX)	78
		Mounting the DB2 CD-ROM (AIX)	80

Copying the contents of the DB2 product CD-ROM to your computer	80	Enabling the execution of remote commands (UNIX).	90
Installing a database partition server on the primary computer using the DB2 Setup wizard (UNIX)	81	Enabling Control Center administration (UNIX).	91
Installing database partition servers on participating computers using a response file (UNIX)	86	Applying the latest FixPak	92
Updating the node configuration file (UNIX).	87	Verifying a partitioned database server installation (UNIX).	93
Enabling communications between database partition servers	89	Installing DB2 online documentation (UNIX).	94

The sections that follow describe how to install DB2 servers in single-partition and multipartition environments on both Windows-based platforms and the AIX platform.

On Windows-based platforms, both DB2 Workgroup Server Edition and DB2 Enterprise Server Edition can function as satellites when they are installed with the Satellite Synchronization component.

When DB2 Enterprise Server Edition is installed with the Satellite Control Server component, it can function as the satellite control server for the satellite environment. The satellite control server can be installed on Windows-based platforms, and on the AIX platform, and can be either a single-partition database environment, or a multipartition database environment.

Installing Single-Partition Database Environments on Windows-Based Platforms

The sections that follow describe how to install single-partition database environments on Windows-based platforms

In a satellite environment, the DB2 Workgroup Server Edition and the DB2 Enterprise Server Edition products can function as satellites when they are installed with the Satellite Synchronization component. The DB2 Enterprise Server Edition product can function as the satellite control server when it is installed with the Satellite Control Server component.

Installation overview for DB2 servers (Windows)

This topic provides an installation overview for DB2[®] Enterprise Server Edition (single-partition) and DB2 Workgroup Server Edition on Windows.

Installation overview:

Preparing your environment for installation

Before you install, you must prepare your computer for installation. To prepare your computer, you will:

1. Verify that your computer meets the necessary installation requirements.
2. Ensure that your system has enough memory to run DB2.
3. Ensure that your system has enough disk space for a DB2 installation.
4. Ensure that you have the necessary user accounts for installation and setup. You require one user account for installation and two user accounts for setup. The user accounts required for setup can be created before you install or you can have the DB2 Setup wizard create them for you.
5. If you are installing on Windows® 2000 and are planning to use Lightweight Directory Access Protocol (LDAP) to register the DB2 server in LDAP, you will extend the Windows 2000 directory schema so that it can contain DB2 object classes and attribute definitions.

Installing DB2

After preparing your environment, you will install DB2 using the DB2 Setup wizard. DB2 Setup wizard features include:

- A DB2 Setup Launchpad from which you can view installation notes, release notes, and learn about DB2 version 8 features.
- Typical, Compact, and Custom installation types.
- The option to install support for multiple languages
- DB2 Administration Server setup (including DAS user setup)
- Administration contact and health monitor notification setup
- Instance setup and configuration (including instance user setup)
- DB2 tools metadata and data warehouse control database setup.
- Response file creation

Some of these tasks can be deferred until after installation, and performed without using the DB2 Setup wizard. For more information on performing these tasks after installation, see the Related information below.

Recommended: Applying the latest FixPak

After you install DB2 using the DB2 Setup wizard, it is recommended that you apply the latest DB2 version 8 FixPak. DB2 FixPaks are available on the IBM® support site.

Verifying the installation

After you install DB2 using the DB2 Setup wizard and have applied the latest DB2 FixPak, it is recommended that you verify the installation. To verify the installation, you will:

1. Create a sample database using the **db2sampl** command. You can also create a sample database using the First Steps utility, if you choose to install it.
2. Once the sample database has been created successfully, you will run SQL commands to retrieve sample data.

Related concepts:

- “Instance creation” in the *Administration Guide: Implementation*

Related tasks:

- “Initializing a warehouse control database during installation” in the *Data Warehouse Center Administration Guide*
- “Tools catalog database and DAS scheduler setup and configuration” in the *Administration Guide: Implementation*
- “Notification and contact list setup and configuration.” in the *Administration Guide: Implementation*

Related reference:

- “Installation requirements for DB2 servers (Windows)” on page 17
- “UPDATE HEALTH NOTIFICATION CONTACT LIST Command” in the *Command Reference*

Installing a DB2 server on Windows

This topic outlines the steps for installing a DB2 Enterprise Server Edition or Workgroup Server Edition single-partition database server on Windows.

Prerequisites:

Ensure that your computer meets the following requirements:

- Installation requirements for DB2 servers
- Memory requirements for DB2 servers
- Disk requirements for DB2 servers
- User accounts for installation and setup of DB2 servers

See the Related references for more information.

Procedure:

It is recommended that you read the Installation overview for DB2 servers prior to beginning the installation.

To install DB2 Enterprise Server Edition or Workgroup Server Edition on Windows:

1. If you are installing on Windows 2000 or Windows .NET and intend to use Lightweight Directory Access Protocol (LDAP) to register the DB2 server in the Active Directory, you must extend the directory schema.
2. Start the DB2 Setup wizard.
3. *Optional:* Apply the latest FixPak.
4. *Optional:* Verify the installation using the Command Line Processor (CLP).
5. *Optional:* Install the DB2 online documentation.

Related concepts:

- “Installation overview for DB2 servers (Windows)” on page 12

Related tasks:

- “Extending the directory schema (Windows 2000 and Windows .NET)” on page 20
- “Starting the DB2 Setup wizard for a DB2 server installation (Windows)” on page 23
- “Applying the latest FixPak” on page 25
- “Verifying the installation using the command line processor (CLP)” on page 26
- “Installing DB2 online documentation (Windows)” on page 27
- “Tools catalog database and DAS scheduler setup and configuration” in the *Administration Guide: Implementation*
- “Notification and contact list setup and configuration.” in the *Administration Guide: Implementation*

Related reference:

- “UPDATE ADMIN CONFIGURATION Command” in the *Command Reference*
- “Installation requirements for DB2 servers (Windows)” on page 17
- “User accounts required for installation of DB2 servers (Windows)” on page 21
- “Memory requirements for DB2 servers (Windows)” on page 19
- “Disk requirements for DB2 servers (Windows)” on page 19

Installation overview for DB2 servers (Windows)

This topic provides an installation overview for DB2® Enterprise Server Edition and DB2 Workgroup Server Edition on Windows® (single-partition). You can use either DB2 product as a satellite. You can also use DB2 Enterprise Server Edition as a satellite control server.

Installation overview:

Preparing your environment for installation

Before you install, you must prepare your computer for installation. To prepare your computer, you will:

1. Verify that your computer meets the necessary installation requirements.
2. Ensure that your system has enough memory to run DB2.
3. Ensure that your system has enough disk space for a DB2 installation.
4. Ensure that you have the necessary user accounts for installation and setup. You require one user account for installation and two user accounts for setup. The user accounts required for setup can be created before you install or you can have the DB2 Setup wizard create them for you.
5. If you are installing on Windows 2000 and are planning to use Lightweight Directory Access Protocol (LDAP) to register the DB2 server in LDAP, you will extend the Windows 2000 directory schema so that it can contain DB2 object classes and attribute definitions.

Installing DB2

After preparing your environment, you will install DB2 using the DB2 Setup wizard. DB2 Setup wizard features include:

- A DB2 Setup Launchpad from which you can view installation notes, release notes, and learn about DB2 version 8 features.
- Typical, Compact, and Custom installation types.
- The option to install support for multiple languages
- DB2 Administration Server setup (including DAS user setup)
- Administration contact and health monitor notification setup
- Instance setup and configuration (including instance user setup)
- DB2 tools metadata and data warehouse control database setup.
- Response file creation

Some of these tasks can be deferred until after installation, and performed without using the DB2 Setup wizard. For more information on performing these tasks after installation, see the Related information below.

Recommended: **Applying the latest FixPak**

After you install DB2 using the DB2 Setup wizard, it is recommended that you apply the latest DB2 version 8 FixPak. DB2 FixPaks are available on the IBM® support site.

Verifying the installation

After you install DB2 using the DB2 Setup wizard and have applied

the latest DB2 FixPak, it is recommended that you verify the installation. To verify the installation, you will:

1. Create a sample database using the **db2sampl** command. You can also create a sample database using the First Steps utility, if you choose to install it.
2. Once the sample database has been created successfully, you will run SQL commands to retrieve sample data.

Installation requirements for DB2 servers (Windows)

To install DB2, the following operating system, software, and communications requirements must be met:

Operating system requirements

DB2 Workgroup Server Edition runs on:

- Windows NT Version 4 with Service Pack 6a or higher
- Windows 2000. Service Pack 2 is required for Windows Terminal Server.
- Windows XP (32-bit)
- Windows .NET (32-bit)

DB2 Enterprise Server Edition runs on:

- Windows NT Version 4 with Service Pack 6a or higher
- Windows 2000. Service Pack 2 is required for Windows Terminal Server.
- Windows .NET (32-bit and 64-bit)

Hardware requirements

For 32-bit DB2 products, a Pentium or Pentium compatible CPU is required. For 64-bit DB2 products, an Itanium or Itanium compatible CPU is required.

Software requirements

- If you plan to use the Tivoli Storage Manager facilities for backup and restore of your databases, you require the Tivoli Storage Manager Client Version 4.2.0 or later. If you are running in a 64-bit environment, you will need Tivoli Storage Manager Client Version 5.1 or later.
- Java Runtime Environment (JRE) Version 1.3.1 is required to run DB2 servers and DB2's Java-based tools, such as the Control Center. The DB2 Setup wizard will install the Java Runtime Environment (JRE) Version 1.3.1 if you choose to install DB2 Java-based tools.
- A browser is required to view online help.

Communication requirements

You can use APPC, TCP/IP, MPTN (APPC over TCP/IP), Named

Pipes, and NetBIOS. To remotely administer a Version 8 DB2 database, you must connect using TCP/IP. DB2 Version 8 servers, using the DB2 Connect server support feature, support only outbound client APPC requests; there is no support for inbound client APPC requests.

- For TCP/IP, Named Pipes, and NetBIOS connectivity, no additional software is required.
- For APPC (CPI-C) connectivity, through the DB2 Connect server support feature, one of the following communication products is required:

Table 1. Supported SNA (APPC) products

Operating system	SNA (APPC) communication product
Windows NT	<ul style="list-style-type: none"> – IBM Communications Server Version 6.1.1 or later – IBM Personal Communications for Windows Version 5.0 with CSD 3 – Microsoft SNA Server Version 3 Service Pack 3 or later
Windows 2000	<ul style="list-style-type: none"> – IBM Communications Server Version 6.1.1 or later – IBM Personal Communications for Windows Version 5.0 with CSD 3 – Microsoft SNA Server Version 4 Service Pack 3 or later
Windows XP	<ul style="list-style-type: none"> – IBM Personal Communications for Windows Version 5.5 with APAR IC23490
Windows .NET	Not supported.

- If you plan to use LDAP (Lightweight Directory Access Protocol), you require either a Microsoft LDAP client or an IBM SecureWay LDAP client V3.1.1.
- If you plan to use the Simple Network Management Protocol (SNMP) subagent, you require DPI 2.0 provided by IBM SystemView Agent. SNMP is not supported with DB2 offerings on Windows 64-bit platforms.

Windows (64-bit) considerations

- Local 32-bit applications are supported.
- 32-bit UDFs and stored procedures are supported.
- SQL requests from remote 32-bit downlevel clients are supported.

- DB2 Version 8 Windows 64-bit servers support connections from DB2 Version 6 and Version 7 32-bit clients only for SQL requests. Connections from Version 7 64-bit clients are not supported.

Windows 2000 Terminal Server installation limitation:

You cannot install DB2 Version 8 from a network mapped drive using a remote session on Windows 2000 Terminal Server edition. The available workaround is to use Universal Naming Convention (UNC) paths to launch the installation, or run the install from the console session.

For example, if the directory `c:\pathA\pathB\...\pathN` on a serverA is shared as `serverdir`, you can open `\\serverA\serverdir\filename.ext` to access the file `c:\pathA\pathB\...\pathN\filename.ext` on server.

Related tasks:

- “Installing a DB2 server on Windows” on page 14

Memory requirements for DB2 servers (Windows)

At a minimum, DB2 requires 256 MB of RAM. Additional memory may be required.

When determining memory requirements, be aware of the following:

- Additional memory may be required for non-DB2 software that may be running on your system.
- Additional memory is required to support database clients.
- Specific performance requirements may determine the amount of memory needed.
- Memory requirements will be affected by the size and complexity of your database system.
- Memory requirements will be affected by the extent of database activity and the number of clients accessing your system.

Related tasks:

- “Installing a DB2 server on Windows” on page 14

Disk requirements for DB2 servers (Windows)

The disk space required for DB2 Enterprise Server Edition (ESE) (single partition) or Workgroup Server Edition (WSE) depends on the type of installation you choose. The DB2 Setup wizard provides Typical, Compact, and Custom installation types. This table provides approximate disk space requirements for each installation type.

Table 2. DB2 ESE (single partition) and DB2 WSE disk requirements

Installation type	Minimum disk space
Typical	350 MB
Compact	100 MB
Custom	100 MB

Exact disk space requirements depend on the features installed and the type of disk drive. You may require significantly more space on FAT drives with large cluster sizes.

Typical installation

DB2 is installed with most features and functionality, using a typical configuration. Typical installation includes graphical tools such as the Control Center and Configuration Assistant. You can also choose to install a typical set of data warehousing or satellite features.

Compact installation

Only the basic DB2 features and functions are installed. Compact installation does not include graphical tools.

Custom installation

A custom installation allows you to select the features you want to install.

The DB2 Setup wizard will provide a disk space estimate for the installation options you select.

Remember to include disk space allowance for required software, communication products, and documentation. In DB2 Version 8, HTML and PDF documentation is provided on separate CD-ROMs.

Related tasks:

- “Installing a DB2 server on Windows” on page 14

Extending the directory schema (Windows 2000 and Windows .NET)

If you plan to use LDAP with Windows 2000 or Windows .NET, you must extend the directory schema to contain DB2 object classes and attribute definitions. You must do this once before you install any DB2 products.

Prerequisites:

Your Windows user account must have Schema Administration authority.

Procedure:

To extend the directory schema:

1. Logon to a domain controller.
2. Run the **db2schex.exe** program from the installation CD with Schema Administration authority. You can run this program with Schema Administration authority, without logging off and logging on again, as follows:

```
runas /user:MyDomain\Administrator x:\db2\Windows\utilities\db2schex.exe
```

where x: represents the CD-ROM letter.

When **db2schex.exe** completes, you can continue with the installation.

Related reference:

- “Installation requirements for DB2 servers (Windows)” on page 17

User accounts required for installation of DB2 servers (Windows)

If you are installing on Windows NT, Windows 2000, Windows XP, or Windows .NET, you require an installation user account and two user accounts for setup. The installation user account must be defined prior to running the DB2 Setup wizard. The setup user accounts (DB2 Administration Server user, and DB2 instance user) can be defined prior to installation or you can have the DB2 Setup program create them for you.

All user account names must adhere to your system naming rules and to DB2 naming rules.

DB2 server user accounts:

Installation user account

A local or domain user account is required to perform the installation. The user account must belong to the *Administrators* group on the machine where you will perform the installation and must have the following user rights:

- *Act as part of the operating system*

You can perform the installation without these user rights, but the installation program will be unable to validate accounts.

DB2 Administration Server user account

A local or domain user account is required for the DB2 Administration Server (DAS). You can create the DAS user account before installing DB2 or you can have the DB2 Setup wizard create it for you. If you want to have the DB2 Setup wizard create a new domain user account, the user account you use to perform the installation must have authority to create domain user accounts. The user account must

belong to the *Administrators* group on the machine where you will perform the installation. This account will be granted the following user rights:

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

The DB2 Administration Server (DAS) is a special DB2 administration service used to support the GUI tools and assist with administration tasks on local and remote DB2 servers. The DAS has an assigned user account that is used to log the DAS service on to the computer when the DAS service is started. It is recommended that the DAS user have SYSADM authority on each of the DB2 systems within your environment so that it can start or stop other instances if required. By default, any user that is part of the *Administrator* group has SYSADM authority.

DB2 instance user account

A local or domain user account is required for the DB2 instance. You can create the DB2 instance user account before installing DB2 or you can have the DB2 Setup wizard create it for you. If you want to have the DB2 Setup wizard create a new domain user account, the user account you use to perform the installation must have authority to create domain user accounts. The user account must belong to the *Administrators* group on the machine where you will perform the installation. This account will be granted the following user rights:

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

Every DB2 instance has one user that is assigned when the instance is created. DB2 logs on with this user name when the instance is started.

Related concepts:

- Appendix G, “User, userID and group naming rules” on page 379

Related tasks:

- “Installing a DB2 server on Windows” on page 14

Starting the DB2 Setup wizard for a DB2 server installation (Windows)

This task describes how to start the DB2 Setup wizard on Windows. You will use the DB2 Setup wizard to define your installation and install DB2 to your system.

Prerequisites:

Before you start the DB2 Setup wizard:

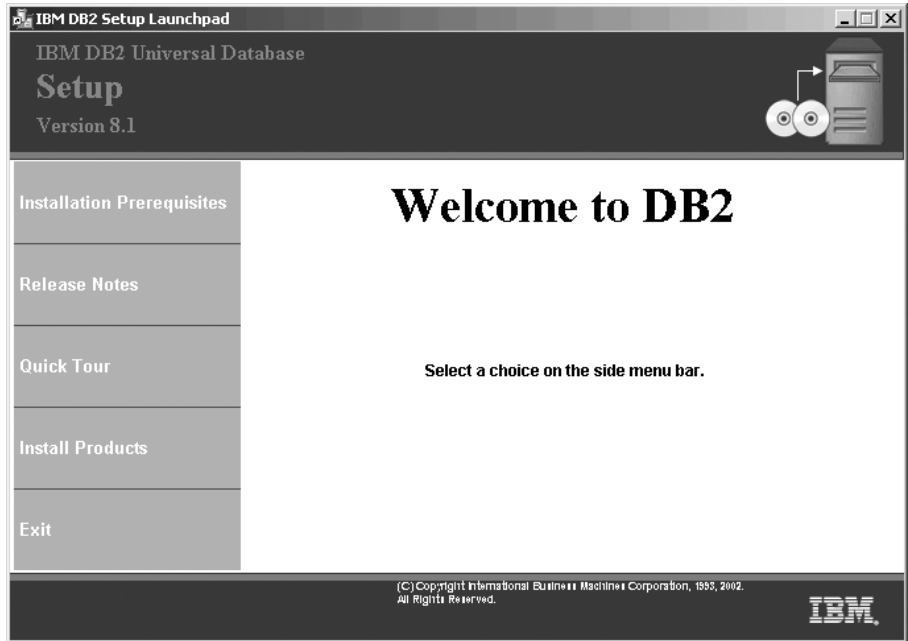
- Ensure that your system meets installation, memory, and disk requirements.
- If you are planning to use LDAP on Windows 2000 or Windows .NET to register the DB2 server in Active Directory, you must extend the directory schema before you install.
- You must have a local *Administrator* user account with the recommended user rights to perform the installation.

Procedure:

To start the DB2 Setup wizard:

1. Log on to the system with the Administrator account that you have defined for DB2 installation.
2. Close all programs so the installation program can update files as required.

3. Insert the CD-ROM into the drive. If enabled, the auto-run feature automatically starts the DB2 Setup launchpad:



From this window, you can view installation prerequisites and the release notes, you can take the DB2 Quick Tour to explore the features of DB2 Universal Database Version 8, or you can proceed directly to the installation. You may want to review the installation prerequisites and release notes for late-breaking information. Select **Install Products** and select the DB2 product to install.

4. The DB2 Setup wizard will determine the system language, and launch the setup program for that language. If you want to run the setup program in a different language, or the setup program failed to auto-start, you can start the DB2 Setup wizard manually.

To start the DB2 Setup wizard manually:

- a. Click **Start** and select the **Run** option.
- b. In the **Open** field, enter the following command:

```
x:\setup /i language
```

where:

- *x*: represents your CD-ROM drive
- *language* is the territory identifier for your language (for example, EN for English).

If the `/i` flag is not specified, the installation program will run in the default language of the operating system.

- c. Click **OK**.
5. Once you have initiated the installation, proceed by following the setup program's prompts. Online help is available to guide you through the remaining steps. To invoke the online help, click Help or press F1. You can click **Cancel** at any time to end the installation. DB2 files will only be copied to your computer once you have clicked **Finish** on the last DB2 Setup wizard installation panel.

If you want to verify your installation using the sample database, be sure to install the sample database component under the Getting Started component group. The sample database is included as part of a Typical installation.

For information on errors encountered during installation, see the db2.log file. The db2.log file stores general information and error messages resulting from the install and uninstall activities. By default, the db2.log file is located in the 'My Documents'\DB2LOG\ directory. The location of the 'My Documents' directory will depend on the settings on your computer.

Related tasks:

- "Installing DB2 Personal Edition (Windows)" on page 97
- "Installing database partition servers on participating computers (Windows)" on page 45
- "Tools catalog database and DAS scheduler setup and configuration" in the *Administration Guide: Implementation*
- "Notification and contact list setup and configuration." in the *Administration Guide: Implementation*

Related reference:

- "UPDATE ADMIN CONFIGURATION Command" in the *Command Reference*
- "Installation requirements for DB2 servers (Windows)" on page 17
- "Language identifiers (for running the DB2 Setup wizard in another language)" in the *Quick Beginnings for DB2 Servers*
- "Memory requirements for DB2 servers (Windows)" on page 19
- "Disk requirements for DB2 servers (Windows)" on page 19

Applying the latest FixPak

Applying the latest FixPak is optionally part of the larger task of installing DB2 products.

A DB2 FixPak contains updates and fixes for bugs (Authorized Program Analysis Reports, or "APARs") found during testing at IBM, as well as fixes

for bugs reported by customers. Every FixPak is accompanied by a document, called APARLIST.TXT, that describes the bug fixes it contains.

FixPaks are cumulative. This means that the latest FixPak for any given version of DB2 contains all of the updates from previous FixPaks for the same version of DB2. We recommend that you keep your DB2 environment running at the latest FixPak level to ensure problem-free operation.

When installing a FixPak on a partitioned ESE system, all participating computers must have the same FixPak installed while the system is offline.

Prerequisites:

Each FixPak may have specific prerequisites. See the FixPak README that accompanies the FixPak for more information.

Procedure:

1. Download the latest DB2 FixPak from the IBM DB2 UDB and DB2 Connect Online Support Web site at <http://www.ibm.com/software/data/db2/udb/winos2unix/support>.
2. Each FixPak contains a set of Release Notes and a README. The README provides instructions for installing the FixPak.

Verifying the installation using the command line processor (CLP)

Verifying the installation using the command line processor (CLP) is optionally part of the larger task of *Installing DB2*.

Once you have completed installing DB2, you can verify the installation by creating a sample database and running SQL commands to retrieve sample data.

Prerequisites:

- The Sample Database component must be installed on your system. The Sample Database component is included in a typical installation.
- You require a user with SYSADM authority.

Procedure:

To verify the installation:

1. Log on to the system as a user with SYSADM authority.
2. Enter the **db2sampl** command to create the SAMPLE database.

This command may take a few minutes to process. There is no completion message; when the command prompt returns, the process is complete.

The SAMPLE database is automatically cataloged with the database alias SAMPLE when it is created.

3. Start the database manager by entering the **db2start** command.
4. Enter the following DB2 commands from a DB2 command window to connect to the SAMPLE database, retrieve a list of all the employees that work in department 20, and reset the database connection:

```
db2 connect to sample
db2 "select * from staff where dept = 20"
db2 connect reset
```

After you have verified the installation, you can remove the SAMPLE database to free up disk space. Enter the **db2 drop database sample** command to drop the SAMPLE database.

Related tasks:

- “Verifying the installation of DB2 servers using First Steps” in the *Quick Beginnings for DB2 Servers*

Installing DB2 online documentation (Windows)

This task describes how to install the DB2 online documentation using the DB2 Setup wizard on Windows. The DB2 online documentation is installed separately from other DB2 products from its own CD-ROM.

Prerequisites:

Before you start the DB2 Setup wizard:

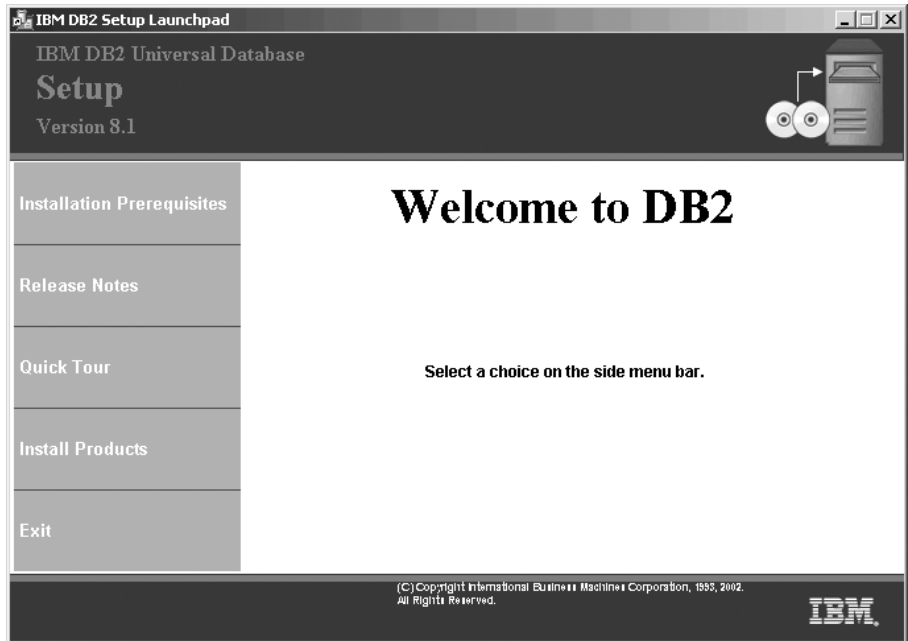
- Ensure that your system meets installation, memory, and disk requirements.
- You must have a local *Administrator* user account with the recommended user rights to perform the installation.

Procedure:

To start the DB2 Setup wizard:

1. Insert the CD-ROM into the drive. The auto-run feature automatically starts the DB2 Setup wizard. The DB2 Setup wizard will determine the system language, and launch the setup program for that language. If you want to run the setup program in a different language, or the setup program failed to auto-start, you can start the DB2 Setup wizard manually.

2. The DB2 Setup Launchpad opens.



From this window, you can view installation prerequisites and the release notes, you can take a Quick Tour to explore the features of DB2 Universal Database Version 8, or you can proceed directly to the installation. You may want to review the installation prerequisites and release notes for late-breaking information.

3. Once you have initiated the installation, proceed by following the setup program's prompts. Online help is available to guide you through the remaining steps. To invoke the online help, click Help or press F1. You can click **Cancel** at any time to end the installation. DB2 files will only be copied to your system once you have clicked **Finish** on the last DB2 Setup wizard installation panel.

For information on errors encountered during installation, see the `db2.log` file. The `db2.log` file stores general information and error messages resulting from the install and uninstall activities. By default, the `db2.log` file is located in the 'My Documents'\DB2LOG\ directory. The location of the 'My Documents' directory will depend on the settings on your computer.

To start the DB2 Setup wizard manually:

1. Click **Start** and select the **Run** option.
2. In the **Open** field, enter the following command:

```
x:\setup /i language
```

where:

- *x*: represents your CD-ROM drive
- *language* is the territory identifier for your language (for example, EN for English).

The /i language parameter is optional. If it is not specified, the DB2 Setup wizard will run in the same language as your operating system.

3. Click **OK**.

Installing Partitioned Database Environments on Windows-Based Platforms

The sections that follow describe how to install multipartition database environments on Windows-based platforms

In a satellite environment, the DB2 Enterprise Server Edition product can function as the satellite control server when it is installed with the Satellite Control Server component. The satellite control server can be a partitioned database environment.

Installing a partitioned DB2 server (Windows)

This topic outlines the steps for installing a partitioned DB2 Enterprise Server Edition database server on Windows.

Prerequisites:

Ensure that your computer meets the following requirements:

1. Installation requirements for partitioned DB2 servers
2. Memory requirements for partitioned DB2 servers
3. Disk requirements for partitioned DB2 servers
4. User accounts for installation and setup of DB2 servers

See the Related references for more information.

Procedure:

It is recommended that you read the Installation overview for partitioned DB2 servers prior to beginning the installation.

To install a partitioned DB2 server:

1. On Windows NT, install Service Pack 6a or higher. On Windows 2000, if you are using Windows Terminal Server, install Service Pack 2 or higher.
2. Prepare the environment for a partitioned DB2 ESE installation.

3. If you are installing on Windows 2000 or Windows .NET and intend to use Lightweight Directory Access Protocol (LDAP) to register the DB2 server in the Active Directory, you must extend the directory schema.
4. Install the instance owning database partition server.
5. Verify port range availability on participating computers.
6. Install database partition servers on participating computers using a response file.
7. *Optional:* Apply the latest FixPak.
8. *Optional:* Verify the partitioned database server installation.
9. *Optional:* Install the DB2 online documentation.

Related concepts:

- “Installation overview for partitioned DB2 servers (Windows)” on page 30

Related tasks:

- “Preparing the environment for a partitioned DB2 server (Windows)” on page 37
- “Extending the directory schema (Windows 2000 and Windows .NET)” on page 20
- “Installing the instance owning database partition server (Windows)” on page 40
- “Verifying port range availability on participating computers” on page 44
- “Installing database partition servers on participating computers (Windows)” on page 45
- “Applying the latest FixPak” on page 25
- “Verifying a partition database server installation (Windows)” on page 49
- “Installing DB2 online documentation (Windows)” on page 27

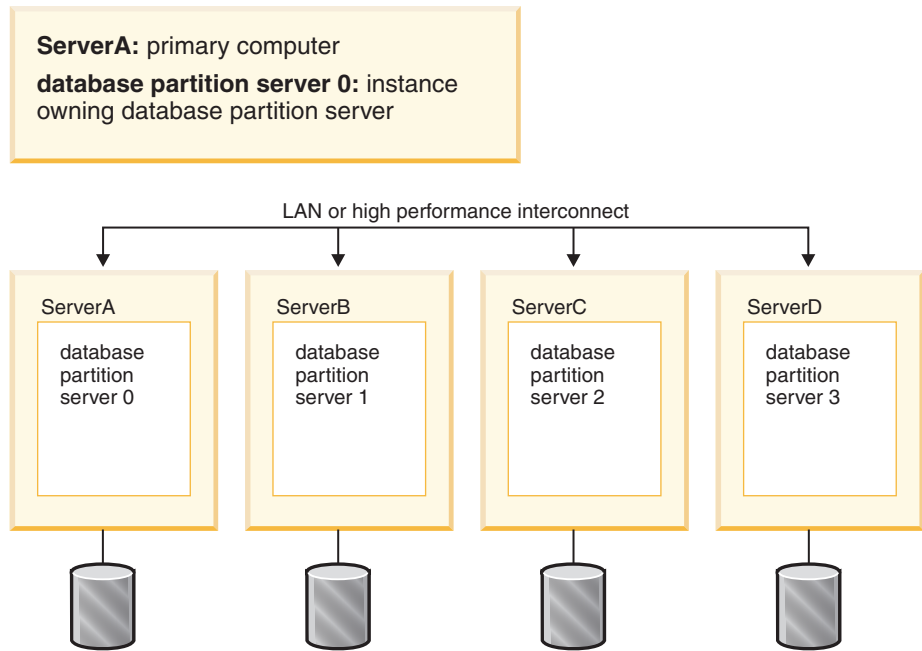
Related reference:

- “User accounts required for installation of DB2 servers (Windows)” on page 21
- “Disk requirements for a partitioned DB2 server (Windows)” on page 36
- “Installation requirements for a partitioned DB2 server (Windows)” on page 33
- “Memory requirements for a partitioned DB2 server (Windows)” on page 35

Installation overview for partitioned DB2 servers (Windows)

The following diagram shows a DB2[®] Enterprise Server Edition (ESE) configuration with four database partition servers, one per computer. Setup instructions are based on this configuration but can easily be adjusted for partitioned configurations with a fewer or greater number of computers and

database partition servers.



ServerA will be referred to as the primary or instance owning computer. ServerB, ServerC, and ServerD will be referred to as participating computers.

Installation overview:

Preparing your environment for installation

Before you install, you must prepare your environment for installation. In some work environments, the System Administrator will perform these tasks. To prepare your environment, you will:

1. Verify that each computer meets the necessary operating system, memory, and disk requirements.
2. Ensure that all computers belong to the same Windows® domain.
3. Ensure that all computers have consistent time and date settings.
4. Verify that all computers can communicate with each other via TCP/IP.
5. Add a domain user account to the local Administrator group on each computer.
6. Optionally create user accounts for setup. The user accounts required for setup can be created before you install or you can have the DB2 Setup wizard create them for you.
7. If you are installing on Windows 2000 or Windows .NET and are planning to use Lightweight Directory Access Protocol (LDAP) to

register your server in the Active Directory, extend the Windows 2000 directory schema to contain DB2 object classes and attribute definitions.

Installing DB2

After preparing your environment, you will install DB2 Enterprise Server Edition:

1. On the primary server (ServerA), you will install an instance-owning database partition server using the DB2 Setup wizard. The DB2 Setup wizard provides the following features:
 - A DB2 Setup Launchpad from which you can view installation notes, release notes, and learn about DB2 version 8 features.
 - Typical, Compact, and Custom installation types.
 - The option to install support for multiple languages
 - DB2 Administration Server setup (including DAS user setup)
 - Administration contact and health monitor notification setup
 - Instance setup and configuration (including instance user setup)
 - DB2 tools metadata and data warehouse control databases setup.
 - Response file creation. You can save your installation choices to a response file for later installation or to duplicate the installation on another computer.

It is recommended that you create a local administration contact list on the instance owning partition. When the DB2 Administration Server is installed and configured on the other participating computers, it will be configured to use the contact list on the instance owning computer.

Some of these task can be deferred until after installation, and performed without using the DB2 Setup wizard. For more information on performing these tasks after installation, see the Related information below.

2. Once you have installed the instance owning database partition server on the primary computer, you will check the port range that DB2 has reserved for database partition communication. You will then ensure that the identical port range is available on each participating computer.
3. Once you have verified that the required port range is available on each participating computer, you will install database partition servers on participating computers using the DB2 Setup wizard.

Verifying the installation

After your system setup is complete, verifying the installation is recommended. To verify the installation, you will:

1. Create a sample database.
2. Run SQL commands to retrieve information from the sample database and ensure that the sample data has been evenly distributed to all database partition servers.

Related concepts:

- “Instance creation” in the *Administration Guide: Implementation*

Related tasks:

- “Initializing a warehouse control database during installation” in the *Data Warehouse Center Administration Guide*
- “Tools catalog database and DAS scheduler setup and configuration” in the *Administration Guide: Implementation*
- “Notification and contact list setup and configuration.” in the *Administration Guide: Implementation*

Related reference:

- “UPDATE HEALTH NOTIFICATION CONTACT LIST Command” in the *Command Reference*

Installation requirements for a partitioned DB2 server (Windows)

This topic lists the installation requirements for a partitioned DB2 server on Windows.

Operating system requirements

DB2 Enterprise Server Edition runs on:

- Windows NT Version 4 with Service Pack 6a or higher (32-bit and 64-bit)
- Windows 2000. Service Pack 2 is required for Windows Terminal Server.
- Windows .NET (32-bit and 64-bit)

Hardware requirements

For 32-bit DB2 products, a Pentium or Pentium compatible CPU is required. For 64-bit DB2 products, an Itanium or Itanium compatible CPU is required.

Software requirements

- If you plan to use the Tivoli Storage Manager facilities for backup and restore of your databases, you require the Tivoli Storage

Manager Client Version 4.2.0 or later. If you are running in a 64-bit environment, you will need Tivoli Storage Manager Client Version 5.1 or later.

- Java Runtime Environment (JRE) Version 1.3.1 is required to run DB2's Java-based tools, such as the Control Center. The DB2 Setup wizard will install the Java Runtime Environment (JRE) Version 1.3.1 if you choose to install DB2 Java-based tools.
- DB2 ESE provides support for host connections.
- A browser is required to view online help.

Communication requirements

You can use TCP/IP, Named Pipes, NetBIOS, and MPTN (APPC over TCP/IP). To remotely administer a Version 8 DB2 database, you must connect using TCP/IP. DB2 Version 8 servers, using the DB2 Connect server support feature, support only outbound client APPC requests; there is no support for inbound client APPC requests.

- For TCP/IP, Named Pipes, and NetBIOS connectivity, no additional software is required.
- For APPC (CPI-C) connectivity, through the DB2 Connect server support feature, one of the following communication products is required:

Table 3. Supported SNA (APPC) products

Operating system	SNA (APPC) communication product
Windows NT	<ul style="list-style-type: none"> – IBM Communications Server Version 6.1.1 or later – IBM Personal Communications for Windows Version 5.0 with CSD 3 – Microsoft SNA Server Version 3 Service Pack 3 or later
Windows 2000	<ul style="list-style-type: none"> – IBM Communications Server Version 6.1.1 or later – IBM Personal Communications for Windows Version 5.0 with CSD 3 – Microsoft SNA Server Version 4 Service Pack 3 or later
Windows .NET	Not supported.

- If you plan to use LDAP (Lightweight Directory Access Protocol), you require either a Microsoft LDAP client or an IBM SecureWay LDAP client V3.1.1.

- If you plan to use the Simple Network Management Protocol (SNMP) subagent, you require DPI 2.0 provided by IBM SystemView Agent. SNMP is not supported with DB2 offerings on Windows 64-bit platforms.

Windows (64-bit) considerations

- Local 32-bit applications are supported.
- 32-bit UDFs and stored procedures are supported.
- SQL requests from remote 32-bit downlevel clients are supported.
- DB2 Version 8 Windows 64-bit servers support connections from DB2 Version 6 and Version 7 32-bit clients only for SQL requests. Connections from Version 7 64-bit clients are not supported.

DB2 Administration Server (DAS) requirements

A DAS must be created on each physical machine for the Control Center and the Task Center to work properly.

Windows 2000 Terminal Server installation limitation

You cannot install DB2 Version 8 from a network mapped drive using a remote session on Windows 2000 Terminal Server edition. The available workaround is to use Universal Naming Convention (UNC) paths to launch the installation, or run the install from the console session.

For example, if the directory `c:\pathA\pathB\...\pathN` on a serverA is shared as `serverdir`, you can open `\\serverA\serverdir\filename.ext` to access the file `c:\pathA\pathB\...\pathN\filename.ext` on server.

Related tasks:

- “Installing a partitioned DB2 server (Windows)” on page 29

Memory requirements for a partitioned DB2 server (Windows)

At a minimum, DB2 requires 256 MB of RAM. Additional memory may be required. In a partitioned database environment, the amount of memory required for each database partition server depends heavily on your configuration.

When determining memory requirements, be aware of the following:

- Additional memory may be required for non-DB2 software that may be running on your system.
- Additional memory is required to support database clients.
- Specific performance requirements may determine the amount of memory needed.

- Memory requirements will be affected by the size and complexity of your database system.
- Memory requirements will be affected by the extent of database activity and the number of clients accessing your system.
- Memory requirements in a partitioned environment may be affected by system design. Demand for memory on one computer may be greater than the demand on another.

Related tasks:

- “Installing a partitioned DB2 server (Windows)” on page 29

Disk requirements for a partitioned DB2 server (Windows)

The disk space required for a DB2 Enterprise Server Edition (ESE) depends on the the type of installation you choose. The DB2 Setup wizard provides Typical, Compact, and Custom installation types. This table provides an approximate disk space requirements for each installation type.

Table 4. DB2 Enterprise Server Edition disk requirements

Installation type	Minimum disk space
Typical	350 MB
Compact	100 MB
Custom	100 MB

Exact disk space requirements depend on the features installed and the type of disk drive. You may require significantly more space on FAT drives with large cluster sizes.

Typical installation

DB2 ESE is installed with most features and functionality, using a typical configuration. Typical installation Includes graphical tools such as the Control Center and Configuration Assistant. You can also choose to install a typical set of data warehousing features.

Compact installation

Only the basic DB2 features and functions are installed. Compact installation does not include graphical tools.

Custom installation

A custom installation allows you to select the features you want to install.

The DB2 Setup wizard will provide a disk space estimate for the installation options you select.

Remember to include disk space allowance for required software, communication products, and documentation. In DB2 Version 8, HTML and PDF documentation is provided on separate CD-ROMs.

Related tasks:

- “Installing a partitioned DB2 server (Windows)” on page 29

Preparing the environment for a partitioned DB2 server (Windows)

This topic describes the steps required to prepare your Windows environment for a partitioned installation of DB2 Enterprise Server Edition.

Restrictions:

Each participating computer must have the same operating system. For example, you cannot have a partitioned database system that includes both Windows NT and Windows 2000 operating systems.

Procedure:

To prepare your Windows environment for installation:

1. Ensure that the primary computer and participating computers belong to the same Windows domain.

Windows NT

Check the domain that computer belongs to using the Network dialog, accessible through the Control Panel.

Windows 2000 or Windows .NET

Check the domain that computer belongs to using the System Properties dialog, accessible through the Control Panel.

2. Ensure that time and date settings on the primary computer and participating computers are consistent. To be considered consistent, the difference in GMT time between all computers must be no greater than 1 hour.

System date and time can be modified using the Date/Time Properties dialog, accessible through the Control Panel. You can use the `max_time_diff` configuration parameter to change this restriction. The default is `max_time_diff = 60`, which allows a difference of less than 60 minutes.

3. Ensure that all participating computers can communicate with each other using TCP/IP:
 - a. On one participating computer, enter the **hostname** command, which will return the hostname of the computer.
 - b. On another participating computer, enter the following command:

ping *hostname*

where *hostname* represents the hostname of the primary computer. If the test is successful, you will receive the output similar to the following:

Pinging ServerA.ibm.com [9.21.27.230] with 32 bytes of data:

Reply from 9.21.27.230: bytes=32 time<10ms TTL=128

Reply from 9.21.27.230: bytes=32 time<10ms TTL=128

Reply from 9.21.27.230: bytes=32 time<10ms TTL=128

Repeat these steps until you are sure that all participating computers can communicate with each other using TCP/IP. Each computer must have a static IP address.

If you are planning to use multiple network adapters, you can specify which adapter to use to communicate between database partition servers. Use the **db2nchg** command to specify the netname field in the `db2nodes.cfg` file after the installation is complete.

4. During the installation you will be asked to provide a local or domain user account that will be used by the DB2 Administration Server (DAS) to log on to the system and to start itself as a service. You can define a user now or have the DB2 Setup wizard create one for you. If you want to create a new domain user using the DB2 Setup wizard, the account used to perform the installation must have authority to create domain users.
5. On the primary computer, where you will install the instance owning partition, you must have a domain user account that belongs to the local *Administrators* group. You must add the same user account to the local *Administrators* group on each participating computer. This user must have the *Act as part of the operating system* user right. You will log on as this user when you install DB2.
6. Ensure that you install DB2 to the same drive on each participating computer. For example, do not install DB2 on the c: drive of the instance owning database server, on the d: drive of a database partition server, and on the j: drive of another database partition server. Install DB2 on the c: drive of instance owning database server and install DB2 on the c: drive of any other participating database partition servers.
7. During the installation you will be asked to provide a domain user account to be associated with the DB2 instance. You can define a user now, or you can have the DB2 Setup wizard create a new domain user for you. If you want to create a new domain user using the DB2 Setup wizard, the account used to perform the installation must have authority to create domain users. The instance user domain account must belong to the local *Administrators* group on all the participating computers and will be granted the following user rights:

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

Related concepts:

- “DB2 system administrator group (Windows)” in the *Quick Beginnings for DB2 Personal Edition*

Related tasks:

- “Granting user rights (Windows)” in the *Quick Beginnings for DB2 Personal Edition*

Related reference:

- “db2nchg - Change Database Partition Server Configuration Command” in the *Command Reference*

Extending the directory schema (Windows 2000 and Windows .NET)

If you plan to use LDAP with Windows 2000 or Windows .NET, you must extend the directory schema to contain DB2 object classes and attribute definitions. You must do this once before you install any DB2 products.

Prerequisites:

Your Windows user account must have Schema Administration authority.

Procedure:

To extend the directory schema:

1. Logon to a domain controller.
2. Run the **db2schex.exe** program from the installation CD with Schema Administration authority. You can run this program with Schema Administration authority, without logging off and logging on again, as follows:

```
runas /user:MyDomain\Administrator x:\db2\Windows\utilities\db2schex.exe
```

where x: represents the CD-ROM letter.

When **db2schex.exe** completes, you can continue with the installation.

Related reference:

- “Installation requirements for DB2 servers (Windows)” on page 17

Installing the instance owning database partition server (Windows)

This task describes how to install the instance owning database partition server on the primary computer using the DB2 Setup wizard.

Prerequisites:

Before you install the instance owning database partition server:

- Ensure that your system meets installation, memory, and disk requirements.
- If you are planning to use LDAP on Windows 2000 or Windows .NET to register the DB2 server in Active Directory, you must extend the directory schema before you install.
- You must have a local *Administrators* user account with the recommended user rights to perform the installation.
- During instance creation a number of ports equal to the number of logical nodes that the instance is capable of supporting will be reserved in the `/etc/services`. These ports will be used by the Fast Communication Manager. The reserved ports will be in the following format:

```
DB2_InstanceName
DB2_InstanceName_1
DB2_InstanceName_2
DB2_InstanceName_END
```

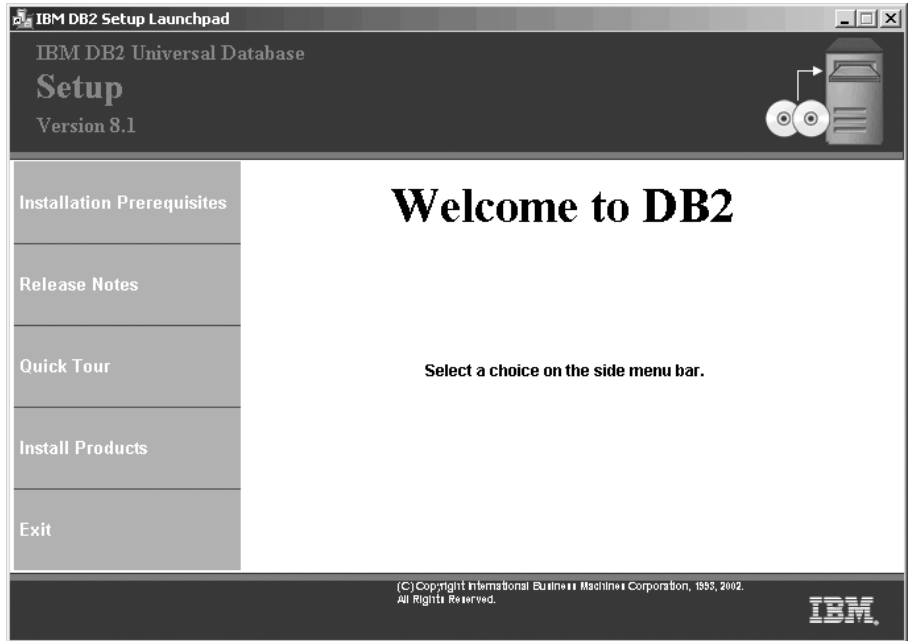
The only mandatory entries are the beginning (DB2_InstanceName) and ending (DB2_InstanceName_END) ports. The other entries are reserved in the services file so that other applications do not use these ports.

Procedure:

To install the instance owning database partition server:

1. Log on to the system with the domain user account that you will use to perform the installation. This is the domain user account that you added to the local *Administrators* group on each computer.
2. Close all programs so the installation program can update files as required.

3. Insert the CD-ROM into the drive. If enabled, the auto-run feature automatically starts the DB2 Setup Launchpad:



From this window, you can view installation prerequisites and the release notes, you can take the DB2 Quick Tour to explore the features of DB2 Universal Database Version 8, or you can proceed directly to the installation. You may want to review the installation prerequisites and release notes for late-breaking information. Select **Install Products** and select the DB2 product to install.

4. The DB2 Setup wizard will determine the system language, and launch the setup program for that language. If you want to run the setup program in a different language, or the setup program failed to auto-start, you can start the DB2 Setup wizard manually.

To start the DB2 Setup wizard manually:

- a. Click **Start** and select the **Run** option.
- b. In the **Open** field, enter the following command:

```
x:\setup /i language
```

where:

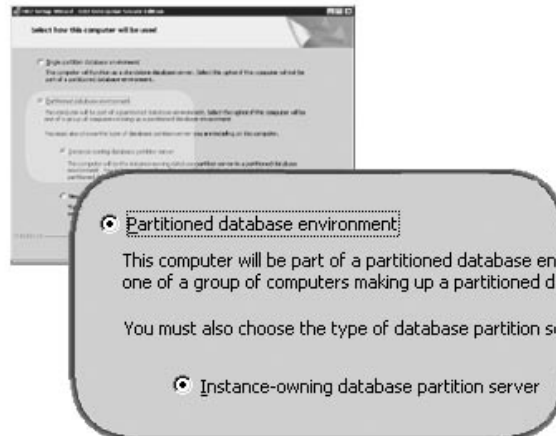
- *x*: represents your CD-ROM drive
- *language* is the territory identifier for your language (for example, EN for English).

If the */i* flag is not specified, the installation program will run in the default language of the operating system.

- c. Click **OK**.
5. When you have finished viewing launchpad information, proceed with the installation. The following list provides information about specific DB2 Setup wizard installation panels and the selections you must make to correctly install the instance owning partition on the primary computer:

Select how this computer will be used

On the *Select how this computer will be used* panel, you must select the **Partitioned database environment** radio button and the **Instance-owning database partition server** radio button.

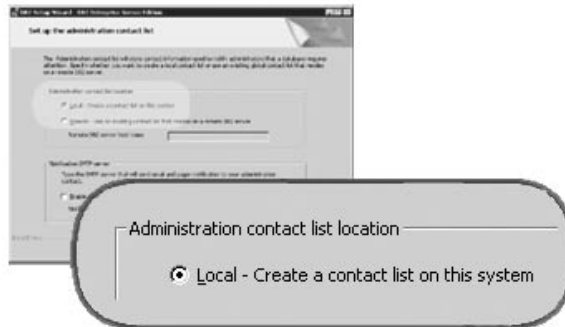


Set up the administration contact list

On the *Set up the administration contact list* panel, select **Local**. This selection will create a file on the primary computer that will store contact information for your system.

The contact information is used by DB2 to send notifications and alerts to a system administrator. A notification may state that a job has completed. An alert may state that a system threshold has been breached. You can specify notification and alert parameters after setup is complete.

Participating computers will remotely access the contact list on this computer.

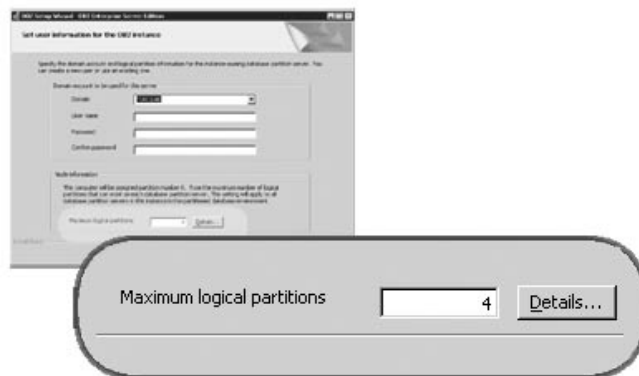


Set user information for the DB2 instance

On the *Set user information for the DB2 instance* panel, you must specify a domain for the DB2 instance and the maximum number of database partitions you can have on a computer.

Select the domain in which your partitioned database will exist from the drop-down box. You can also specify a domain name by entering the domain name in the **Domain** field.

The default maximum logical partitions for a computer is four. If you have one database partition server per computer, only one port is required. If you keep the default value of four, four ports will be reserved for database partition server communication. DB2 will attempt to reserve identical port numbers when you install database partition servers on participating computers.



Online help is available to guide you through the remaining steps. To invoke the online help, click **Help** or press **F1**. You can click **Cancel** at any

time to end the installation. DB2 files will only be copied to your system once you have clicked **Finish** on the last DB2 Setup wizard installation panel.

For information on errors encountered during installation, see the `db2.log` file. The `db2.log` file stores general information and error messages resulting from the install and uninstall activities. By default, the `db2.log` file is located in the 'My Documents'\DB2LOG\ directory. The location of the 'My Documents' directory will depend on the settings on your computer.

Related tasks:

- “Installing database partition servers on participating computers (Windows)” on page 45

Related reference:

- “Language identifiers (for running the DB2 Setup wizard in another language)” in the *Quick Beginnings for DB2 Servers*
- “Disk requirements for a partitioned DB2 server (Windows)” on page 36
- “Installation requirements for a partitioned DB2 server (Windows)” on page 33
- “Memory requirements for a partitioned DB2 server (Windows)” on page 35

Verifying port range availability on participating computers

This task describes the steps required to verify port range availability on participating computers. The port range will be used by the Fast Communications Manager (FCM). FCM is a feature of DB2 that handles communications between database partition servers.

When you install the instance owning database partition server on the primary computer, DB2 reserves a port range according to the specified number of database partition servers per node. The default range is for four ports. The DB2 Setup wizard must be able to reserve an identical port range when database partition servers are installed on participating computers.

Procedure:

To verify port range availability on participating computers:

1. Open the services file located in the `%SystemRoot%\system32\drivers\etc` directory, where `%SystemRoot%` is your Windows root directory.
2. Locate the ports reserved for the DB2 Fast Communications Manager (FCM). The entries should appear similar to the following:

DB2_db2inst1	60000/tcp
DB2_db2inst1_1	60001/tcp
DB2_db2inst1_2	60002/tcp
DB2_db2inst1_END	60003/tcp

DB2 will reserve the first four available ports after 60000.

3. On each participating computer, open the services file and verify that the ports reserved for DB2 FCM in the services file of the primary computer are not being used.
4. In the unlikely event that the required ports are in use on a participating computer, identify an available port range for all computers and update each service file, including the service file on the primary computer.

Related concepts:

- “Fast Communications Manager (Windows)” in the *Quick Beginnings for DB2 Servers*

Related tasks:

- “Installing database partition servers on participating computers (Windows)” on page 45

Related reference:

- “DB2 node configuration file (db2nodes.cfg)” in the *Quick Beginnings for DB2 Servers*

Installing database partition servers on participating computers (Windows)

This task describes how to install database partition servers on participating computers using the DB2 Setup wizard. You must perform this task on each participating computer.

Prerequisites:

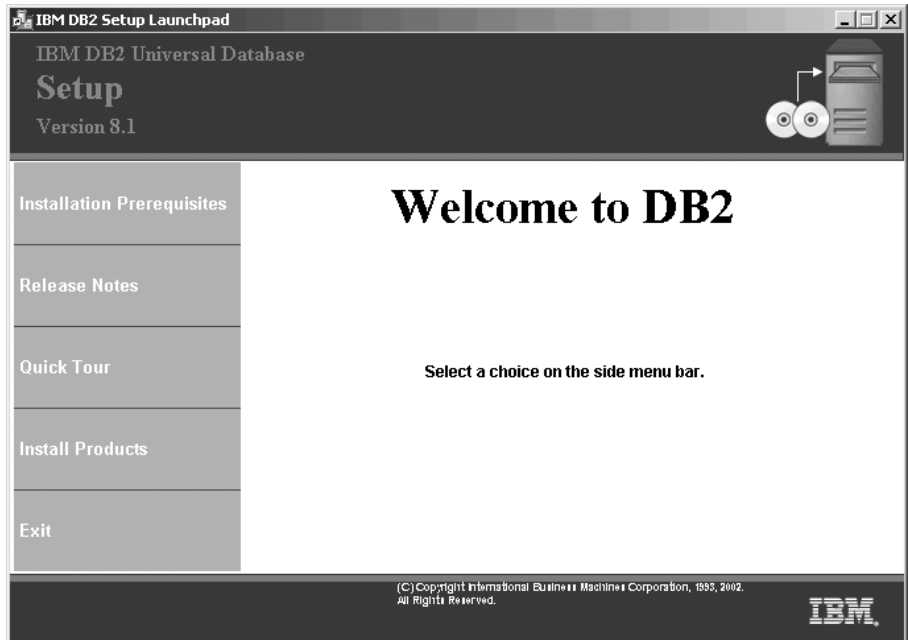
Before you install a database partition server on a participating computer:

- The instance owning database server partition must be installed on the primary computer.
- The domain user account that you added to the local Administrators group on the primary computer, must be added to the local Administrators group on the participating computer. You will use this account to perform the installation.

Procedure:

To start the DB2 Setup wizard:

1. Log on to the system with the domain user account that you will use to perform the installation. This is the domain user account that you added to the local *Administrators* group on each computer.
2. Close all programs so the installation program can update files as required.
3. Insert the CD-ROM into the drive. If enabled, the auto-run feature automatically starts the DB2 Setup Launchpad:



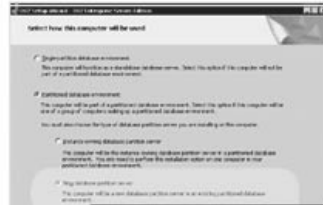
From this window, you can view installation prerequisites and the release notes, you can take the DB2 Quick Tour to explore the features of DB2 Universal Database Version 8, or you can proceed directly to the installation. You may want to review the installation prerequisites and release notes for late-breaking information. Select **Install Products** and select the DB2 product to install.

4. The DB2 Setup wizard will determine the system language, and launch the setup program for that language. If you want to run the setup program in a different language, or the setup program failed to auto-start, you can start the DB2 Setup wizard manually. The syntax for starting the DB2 Setup wizard is described at the end of this procedure.
5. The following list provides information about specific DB2 Setup wizard installation panels and the selections you must make to correctly install a database partition server on a participating computer:

Select how this computer will be used

On the *Select how this computer will be used* panel, you must select

the **Partitioned database environment** radio button and the **New database partition server** radio button.



New database partition server

This computer will be a new database partition server in an existing partitioned database environment.

Set up an administration contact list

On the *Set up an administration contact list panel*, select **Remote**. Specify the hostname of the primary computer where you installed the instance owning database partition server and set up the contact list.



Remote - Use an existing contact list that resides on a remote DB2 server

Remote DB2 server host name

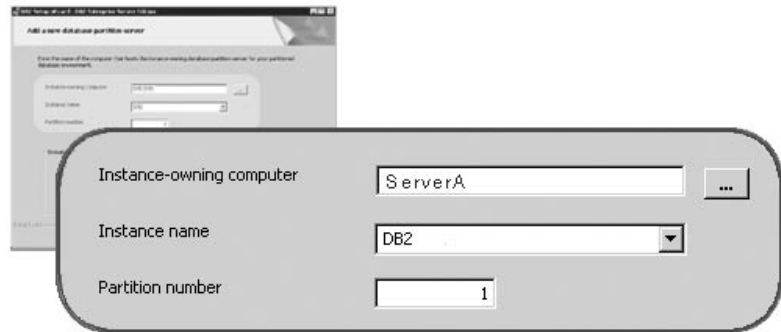
ServerA

Add a new database partition server

On the *Add a new database partition server panel*:

- Specify the hostname of the primary computer (instance-owning computer), where you installed the instance owning database partition server.
- In the drop-down box, select the name of the instance that was created when you installed the instance-owning database partition server. The default name instance name is *DB2*.
- For the partition number, specify a unique value in the range of 1 to 999. If this is the first new database partition server you are installing, it is recommended that you enter a value of 1. For the next database partition server, enter 2, and so on. The instance

owning partition server is always assigned partition number 0. Values must be in ascending order, but gaps in the sequence are acceptable.



Online help is available to guide you through the remaining steps. To invoke the online help, click **Help** or press **F1**. You can click **Cancel** at any time to end the installation. DB2 files will only be copied to your system once you have clicked **Finish** on the last DB2 Setup wizard installation panel.

For information on errors encountered during installation, see the `db2.log` file. The `db2.log` file stores general information and error messages resulting from the install and uninstall activities. By default, the `db2.log` file is located in the 'My Documents'\DB2LOG\ directory. The location of the 'My Documents' directory will depend on the settings on your computer.

To start the DB2 Setup wizard manually:

1. Click **Start** and select the **Run** option.
2. In the **Open** field, enter the following command:

```
x:\setup /i language
```

where:

- *x*: represents your CD-ROM drive
- *language* is the territory identifier for your language (for example, EN for English).

3. Click **OK**.

Applying the latest FixPak

Applying the latest FixPak is optionally part of the larger task of installing DB2 products.

A DB2 FixPak contains updates and fixes for bugs (Authorized Program Analysis Reports, or "APARs") found during testing at IBM, as well as fixes for bugs reported by customers. Every FixPak is accompanied by a document, called APARLIST.TXT, that describes the bug fixes it contains.

FixPaks are cumulative. This means that the latest FixPak for any given version of DB2 contains all of the updates from previous FixPaks for the same version of DB2. We recommend that you keep your DB2 environment running at the latest FixPak level to ensure problem-free operation.

When installing a FixPak on a partitioned ESE system, all participating computers must have the same FixPak installed while the system is offline.

Prerequisites:

Each FixPak may have specific prerequisites. See the FixPak README that accompanies the FixPak for more information.

Procedure:

1. Download the latest DB2 FixPak from the IBM DB2 UDB and DB2 Connect Online Support Web site at <http://www.ibm.com/software/data/db2/udb/winos2unix/support>.
2. Each FixPak contains a set of Release Notes and a README. The README provides instructions for installing the FixPak.

Verifying a partition database server installation (Windows)

To verify that your DB2 server installation was successful, you will create a sample database and run SQL commands to retrieve sample data and to verify that the data has been distributed to all participating database partition servers.

Prerequisites:

You have completed all of the installation steps.

Procedure:

To create the SAMPLE database:

1. Log on to the primary computer (ServerA). as user with SYSADM authority.
2. Enter the **db2sampl** command to create the SAMPLE database.

This command may take a few minutes to process. There is no completion message; when the command prompt returns, the process is complete.

The SAMPLE database is automatically cataloged with the database alias SAMPLE when it is created.

3. Start the database manager by entering the **db2start** command.
4. Enter the following DB2 commands from a DB2 command window to connect to the SAMPLE database, retrieve a list of all the employees that work in department 20:

```
db2 connect to sample
db2 "select * from staff where dept = 20"
```
5. To verify that data has been distributed across database partition servers, enter the following commands from a DB2 command window:

```
select distinct dbpartitionnum(empno) from employee;
```

The output will list the database partitions used by the employee table. The specific output will depend on the number of partitions in the database and the number of partitions in the partition group that is used by the tablespace where the employee table was created.

After you have verified the installation, you can remove the SAMPLE database to free up disk space. Enter the **db2 drop database sample** command to drop the SAMPLE database.

Installing DB2 online documentation (Windows)

This task describes how to install the DB2 online documentation using the DB2 Setup wizard on Windows. The DB2 online documentation is installed separately from other DB2 products from its own CD-ROM.

Prerequisites:

Before you start the DB2 Setup wizard:

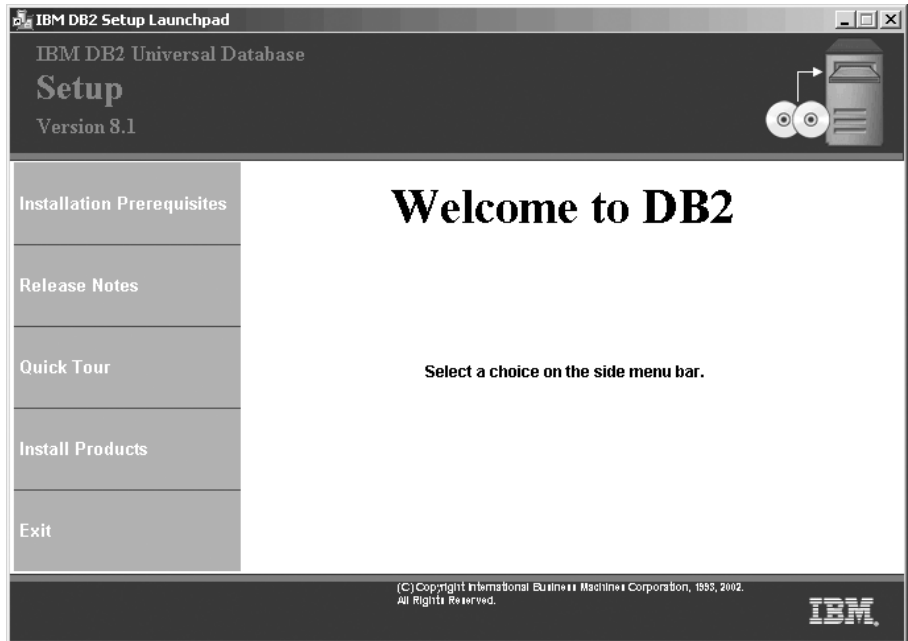
- Ensure that your system meets installation, memory, and disk requirements.
- You must have a local *Administrator* user account with the recommended user rights to perform the installation.

Procedure:

To start the DB2 Setup wizard:

1. Insert the CD-ROM into the drive. The auto-run feature automatically starts the DB2 Setup wizard. The DB2 Setup wizard will determine the system language, and launch the setup program for that language. If you want to run the setup program in a different language, or the setup program failed to auto-start, you can start the DB2 Setup wizard manually.

2. The DB2 Setup Launchpad opens.



From this window, you can view installation prerequisites and the release notes, you can take a Quick Tour to explore the features of DB2 Universal Database Version 8, or you can proceed directly to the installation. You may want to review the installation prerequisites and release notes for late-breaking information.

3. Once you have initiated the installation, proceed by following the setup program's prompts. Online help is available to guide you through the remaining steps. To invoke the online help, click Help or press F1. You can click **Cancel** at any time to end the installation. DB2 files will only be copied to your system once you have clicked **Finish** on the last DB2 Setup wizard installation panel.

For information on errors encountered during installation, see the db2.log file. The db2.log file stores general information and error messages resulting from the install and uninstall activities. By default, the db2.log file is located in the 'My Documents'\DB2LOG\ directory. The location of the 'My Documents' directory will depend on the settings on your computer.

To start the DB2 Setup wizard manually:

1. Click **Start** and select the **Run** option.
2. In the **Open** field, enter the following command:

```
x:\setup /i language
```

where:

- *x*: represents your CD-ROM drive
- *language* is the territory identifier for your language (for example, EN for English).

The */i* language parameter is optional. If it is not specified, the DB2 Setup wizard will run in the same language as your operating system.

3. Click **OK**.

Installing Single-Partition Database Environments on the AIX Platform

The sections that follow describe how to install single-partition database environments on UNIX-based platforms

In a satellite environment, the DB2 Enterprise Server Edition product can function as the satellite control server when it is installed with the Satellite Control Server component on the AIX platform.

Installation overview for DB2 servers (UNIX)

This topic provides an overview of the steps required to install DB2[®] Enterprise Server Edition (single partition) or Workgroup Server Edition on UNIX[®] systems using the DB2 Setup wizard.

Installation overview:

Preparing your environment for installation

Before you install, you must prepare your computer for installation. To prepare your computer, you will:

1. Verify that your computer meets the necessary operating system, memory, and disk requirements.
2. Update kernel parameters to the recommended values (HP-UX, Linux, Solaris Operating Environment). A system restart is required.
3. Mount the installation CD-ROM.

Installing DB2

After preparing your environment, you will install DB2 using the DB2 Setup wizard. DB2 Setup wizard features include:

- A DB2 Setup Launchpad from which you can view installation notes, release notes, and learn about DB2 version 8 features.
- Typical, Compact, and Custom installation types. Installation choices presented to you depend on the type of installation you choose.
- The option to install support for multiple languages

- DB2 Administration Server setup (including DAS user setup)
- Administration contact and health monitor notification setup
- Instance setup and configuration (including instance user setup)
- DB2 tools metadata Setup. Metadata is required for DB2 tools to function.
- Response file creation

Some of these task can be deferred until after installation, and performed without using the DB2 Setup wizard. For more information on performing these tasks after installation, see the Related information below.

Installing the latest FixPak

After you install DB2 using the DB2 Setup wizard, it is recommended that you install the latest DB2 version 8 FixPak. DB2 FixPaks are available on the IBM® support site.

Verifying the installation

After you install DB2 using the DB2 Setup wizard and have applied the latest DB2 FixPak, it is recommended that you verify the installation. To verify the installation, you will:

1. Create a sample database using the **db2samp1** command. You can also create a sample database using the First Steps utility, if you choose to install it.
2. Once the sample database has been created successfully, you can run SQL commands to retrieve sample data.

Related concepts:

- “Instance creation” in the *Administration Guide: Implementation*
- “Installation overview for a partitioned DB2 server (UNIX)” on page 64

Related tasks:

- “Initializing a warehouse control database during installation” in the *Data Warehouse Center Administration Guide*
- “Tools catalog database and DAS scheduler setup and configuration” in the *Administration Guide: Implementation*
- “Notification and contact list setup and configuration.” in the *Administration Guide: Implementation*
- “Installing DB2 servers on AIX” on page 54
- “Installing a DB2 server on HP-UX” in the *Quick Beginnings for DB2 Servers*
- “Installing a DB2 server on Linux” in the *Quick Beginnings for DB2 Servers*
- “Installing a DB2 server on Solaris” in the *Quick Beginnings for DB2 Servers*

Related reference:

- “UPDATE HEALTH NOTIFICATION CONTACT LIST Command” in the *Command Reference*

Installing DB2 servers on AIX

This topic outlines steps for installing DB2 Enterprise Server Edition (single partition) or DB2 Workgroup Server Edition on AIX.

Prerequisites:

Ensure that your computer meets the following requirements:

1. Installation requirements for DB2 servers
2. Memory requirements for DB2 servers
3. Disk requirements for DB2 servers
4. Groups and user accounts for DB2 installations

See the Related references for more information.

Procedure:

It is recommended that you read the Installation overview for DB2 servers prior to beginning the installation.

To install DB2 on AIX:

1. Mount the DB2 installation CD-ROM.
2. Start the DB2 Setup wizard to install DB2.
3. *Optional:* Apply the latest FixPak.
4. *Optional:* Verify the installation using the command line processor (CLP).
5. *Optional:* Install the DB2 online documentation.

Related concepts:

- “Installation overview for DB2 servers (UNIX)” on page 52

Related tasks:

- “Mounting the DB2 CD-ROM (AIX)” on page 58
- “Starting the DB2 Setup wizard for a DB2 server installation (UNIX)” on page 58
- “Applying the latest FixPak” on page 25
- “Verifying the installation using the command line processor (CLP)” on page 26
- “Installing DB2 online documentation (UNIX)” on page 62
- “Creating group and user IDs for a DB2 installation” in the *Installation and Configuration Supplement*

Related reference:

- “Installation requirements for DB2 servers (AIX)” on page 55
- “Memory requirements for servers (UNIX)” on page 56
- “Disk requirements for DB2 servers (UNIX)” on page 57

Installation requirements for DB2 servers (AIX)

This topic lists the hardware, operating system, software, and communications requirements for DB2 Enterprise Server Edition and DB2 Workgroup Server Edition.

Hardware requirements

One of:

- IBM RISC/6000
- eServer pSeries

Operating system requirements

DB2 Enterprise Server Edition is available on:

- AIX Version 4.3.3 with maintenance level 9 or later (32-bit)
- AIX Version 5.1.0 with maintenance level 2 or later (32-bit and 64-bit)

DB2 Workgroup Server Edition is available on:

- AIX Version 4.3.3.0 with maintenance level 9 or later (32-bit)
- AIX Version 5L with maintenance level 2 or later (32-bit)

The following AIX filesets are required to install or run DB2 in languages other than English:

- X11.fnt.ucs.ttf (AIX Windows Unicode TrueType Fonts)
- x1C.rte 5.0.2.x
- For Asian languages, the following filesets are also required:
 - X11.fnt.ucs.ttf_CN (for zh_CN or Zh_CN)
 - X11.fnt.ucs.ttf_KR (for ko_KR)
 - X11.fnt.ucs.ttf_TW (for zh_TW or Zh_TW)
- On AIX Version 4.3.3 the following fileset is required:
 - x1C.aix43.rte 5.0.2.x
- On AIX Version 5L the following fileset is required:
 - x1C.aix50.rte 5.0.2.x

AIX filesets can be downloaded from:

<http://techsupport.services.ibm.com/server/fixes>

Software requirements

- Java Runtime Environment (JRE) Version 1.3.1 is required to run DB2 servers and DB2's Java-based tools, such as the Control Center.
- If you plan to use the Tivoli Storage Manager facilities to back up and restore your databases, you require the Tivoli Storage Manager Client Version 4.2.0 or later.
- A browser is required to view online help.

Communication requirements

You can use APPC, TCP/IP or MPTN (APPC over TCP/IP). DB2 Version 8 servers, using the DB2 Connect server support feature, support only outbound client APPC requests; there is no support for inbound client APPC requests. You can only use TCP/IP to remotely administer databases.

- For TCP/IP connectivity, no additional software is required.
- For APPC (CPI-C) connectivity, through the DB2 Connect server support feature, one of the following communication products is required:
 - IBM eNetwork Communications Server for AIX V5.0.3
 - Bull DPX/20 SNA/20
- For LDAP (Lightweight Directory Access Protocol) support, you require an IBM SecureWay Directory Client V3.1.1
- If you plan to use the Simple Network Management Protocol (SNMP) subagent, you require DPI 2.0 provided by IBM SystemView Agent.

Installing DB2 products or sharing instance directory on NFS

Currently, we do not support the installation of DB2 products on NFS. Installing DB2 on NFS (for example, NFS mounting /usr/opt/db2_08_01 or /opt/IBM/db2/V8.1) can be error prone and these errors can be difficult to diagnose.

The following configuration is not supported:

- Setting up an instance on a filesystem
- NFS mounting a filesystem from multiple machines, and then run DB2 on these machines using that same instance.

This configuration can cause file locking and performance problems.

Related tasks:

- “Installing DB2 servers on AIX” on page 54

Memory requirements for servers (UNIX)

At a minimum, DB2 requires 256 MB of RAM. Additional memory may be required.

When determining memory requirements, be aware of the following:

- Additional memory may be required for non-DB2 software that may be running on your system.
- Additional memory is required to support database clients.
- Specific performance requirements may determine the amount of memory needed.
- Memory requirements will be affected by the size and complexity of your database system.
- Memory requirements will be affected by the extent of database activity and the number of clients accessing your system.

Related tasks:

- “Installing DB2 servers on AIX” on page 54
- “Installing a DB2 server on HP-UX” in the *Quick Beginnings for DB2 Servers*
- “Installing a DB2 server on Linux” in the *Quick Beginnings for DB2 Servers*
- “Installing a DB2 server on Solaris” in the *Quick Beginnings for DB2 Servers*

Disk requirements for DB2 servers (UNIX)

The disk space required for DB2 Enterprise Server Edition or Workgroup Server Edition depends on the type of installation you choose. The DB2 Setup wizard provides typical, compact, and custom installation types. This table provides an approximate disk space requirement for each installation type.

Table 5. DB2 server disk requirements

Installation type	Required disk space
Typical	450 to 550 MB
Compact	350 to 400 MB
Custom	350 to 700 MB

Typical installation

DB2 is installed with most features and functionality, using a typical configuration. Typical installation includes graphical tools such as the Control Center and Configuration Assistant. You can also choose to install a typical set of data warehousing features.

Compact installation

Only the basic DB2 features and functions are installed. Compact installation does not include graphical tools.

Custom installation

A custom installation allows you to select the features you want to install.

The DB2 Setup wizard will provide a disk space estimate for the installation options you select.

Remember to include disk space allowance for required software, communication products, and documentation. In DB2 version 8, HTML and PDF documentation is provided on separate CD-ROMs.

Related tasks:

- “Installing DB2 servers on AIX” on page 54
- “Installing a DB2 server on HP-UX” in the *Quick Beginnings for DB2 Servers*
- “Installing a DB2 server on Linux” in the *Quick Beginnings for DB2 Servers*
- “Installing a DB2 server on Solaris” in the *Quick Beginnings for DB2 Servers*

Mounting the DB2 CD-ROM (AIX)

You must mount the DB2 product CD-ROM before you can start the DB2 Setup wizard.

Procedure:

To mount the DB2 installation CD and copy the contents:

1. Create a directory for the CD-ROM by entering the following command:

```
mkdir /cdrom -p
```

2. Allocate a CD-ROM file system by entering the following command:

```
crfs -v cdrfs -p ro -d cd0 -m /cdrom
```

where `cd0` is the standard representation for the CD-ROM drive.

3. Mount the CD-ROM file system by entering the following command:

```
mount /cdrom
```

Starting the DB2 Setup wizard for a DB2 server installation (UNIX)

This task describes how to start the DB2 Setup wizard on UNIX systems. The DB2 Setup wizard is used to define your installation preferences and install DB2 to your system.

Prerequisites:

Before you start the DB2 Setup wizard

- Ensure that your system meets installation, memory, and disk requirements.
- You require root authority to perform the installation.
- The DB2 product CD-ROM must be mounted on your system.
- The DB2 Setup wizard is a graphical installer. You must have Xwindow software capable of rendering a graphical user interface for the DB2 Setup

wizard to run on your machine. Ensure that you have properly exported your display. For example, export DISPLAY=9.26.163.144:0.

- If NIS/NIS+ or similar security software is used in your environment, you must manually create required DB2 users before you start the DB2 Setup wizard. Refer to the referenced NIS topic before you begin.
- (Solaris Operating Environment only) You need to have a filesystem with 2 GB of free space to contain the tar.Z file and the uncompressed installation image, in addition to the software disk requirements.

Procedure:

To start the DB2 Setup wizard:

1. Log on to the system as a user with root authority.
2. Refer to the CD-ROM label to ensure that you are using the CD-ROM with your appropriate language.
3. Change to the directory where the CD-ROM is mounted by entering the following command:

```
cd /cdrom
```

where */cdrom* represents mount point of the CD-ROM.

4. See the appropriate section for your operation system:

For AIX, HP-UX and Linux

Enter the **./db2setup** command to start the DB2 Setup wizard.

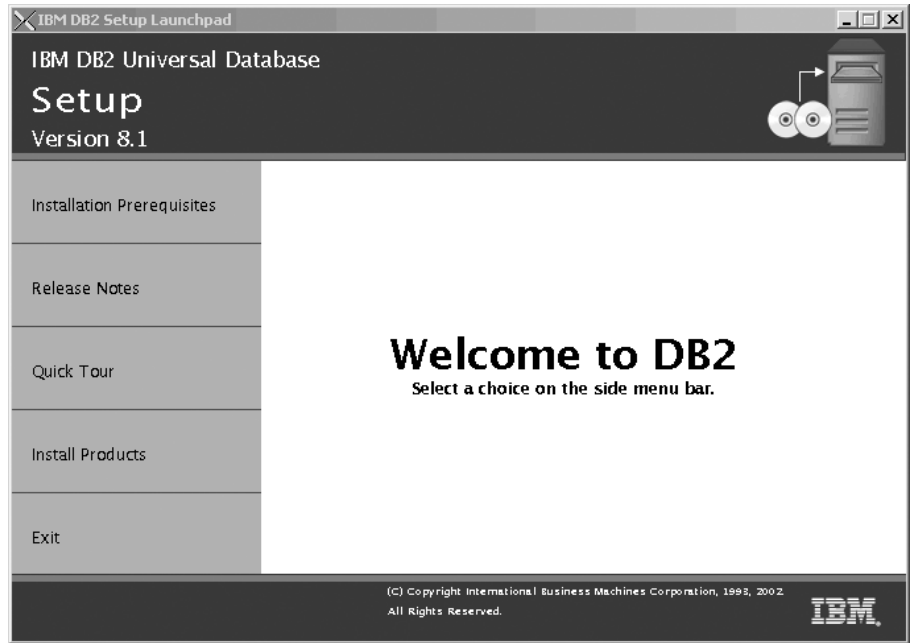
For Solaris Operating Environment

- a. Copy *product.tar.Z*, where *product* represents the product you are licensed to install, to a temporary filesystem.
- b. Enter the following command to start the DB2 Setup wizard:
zcat product.tar.Z | tar -xf - ; ./product/db2setup command

For example, if the product name for DB2 Enterprise Server Edition is *ese*, then enter the following command:

```
zcat ese.tar.Z | tar -xf - ; ./ese/db2setup
```

5. After a moment, the IBM DB2 Setup Launchpad opens.



From this window, you can view installation prerequisites and the release notes, you can take a Quick Tour to explore the features of DB2 Universal Database Version 8, or you can proceed directly to the installation. You may want to review the installation prerequisites and release notes for late-breaking information.

Once you have initiated the installation, proceed through the DB2 Setup wizard installation panels and make your selections. Installation help is available to guide you through the remaining steps. To invoke the installation help, click **Help** or press **F1**. You can click **Cancel** at any time to end the installation. DB2 files will only be copied to your system once you have clicked **Finish** on the last DB2 Setup wizard installation panel.

When you have completed your installation, DB2 will be installed in the one of the following directories:

AIX /usr/opt/db2_08_01

HP-UX, Linux, Solaris Operating Environment
/opt/IBM/db2/V8.1

Related tasks:

- “Tools catalog database and DAS scheduler setup and configuration” in the *Administration Guide: Implementation*

- “Notification and contact list setup and configuration.” in the *Administration Guide: Implementation*

Related reference:

- “UPDATE ADMIN CONFIGURATION Command” in the *Command Reference*
- “db2setup - Install DB2 Command” in the *Command Reference*

Applying the latest FixPak

Applying the latest FixPak is optionally part of the larger task of installing DB2 products.

A DB2 FixPak contains updates and fixes for bugs (Authorized Program Analysis Reports, or “APARs”) found during testing at IBM, as well as fixes for bugs reported by customers. Every FixPak is accompanied by a document, called APARLIST.TXT, that describes the bug fixes it contains.

FixPaks are cumulative. This means that the latest FixPak for any given version of DB2 contains all of the updates from previous FixPaks for the same version of DB2. We recommend that you keep your DB2 environment running at the latest FixPak level to ensure problem-free operation.

When installing a FixPak on a partitioned ESE system, all participating computers must have the same FixPak installed while the system is offline.

Prerequisites:

Each FixPak may have specific prerequisites. See the FixPak README that accompanies the FixPak for more information.

Procedure:

1. Download the latest DB2 FixPak from the IBM DB2 UDB and DB2 Connect Online Support Web site at <http://www.ibm.com/software/data/db2/udb/winos2unix/support>.
2. Each FixPak contains a set of Release Notes and a README. The README provides instructions for installing the FixPak.

Verifying the installation using the command line processor (CLP)

Verifying the installation using the command line processor (CLP) is optionally part of the larger task of *Installing DB2*.

Once you have completed installing DB2, you can verify the installation by creating a sample database and running SQL commands to retrieve sample data.

Prerequisites:

- The Sample Database component must be installed on your system. The Sample Database component is included in a typical installation.
- You require a user with SYSADM authority.

Procedure:

To verify the installation:

1. Log on to the system as a user with SYSADM authority.
2. Enter the **db2sampl** command to create the SAMPLE database.
This command may take a few minutes to process. There is no completion message; when the command prompt returns, the process is complete.
The SAMPLE database is automatically cataloged with the database alias SAMPLE when it is created.
3. Start the database manager by entering the **db2start** command.
4. Enter the following DB2 commands from a DB2 command window to connect to the SAMPLE database, retrieve a list of all the employees that work in department 20, and reset the database connection:

```
db2 connect to sample
db2 "select * from staff where dept = 20"
db2 connect reset
```

After you have verified the installation, you can remove the SAMPLE database to free up disk space. Enter the **db2 drop database sample** command to drop the SAMPLE database.

Related tasks:

- “Verifying the installation of DB2 servers using First Steps” in the *Quick Beginnings for DB2 Servers*

Installing DB2 online documentation (UNIX)

This task describes how to install the DB2 online documentation using the DB2 Setup wizard on UNIX. The DB2 online documentation is installed separately from other DB2 products from its own CD-ROM.

Prerequisites:

Before you start the DB2 Setup wizard

- You require root authority to perform the installation.
- The DB2 product CD-ROM must be mounted on your system.
- The DB2 Setup wizard is a graphical installer. In order for it to run on your machine, you must have Xwindow software capable of rendering a graphical user interface.

- A Java Runtime Environment (JRE) must already be installed.

Procedure:

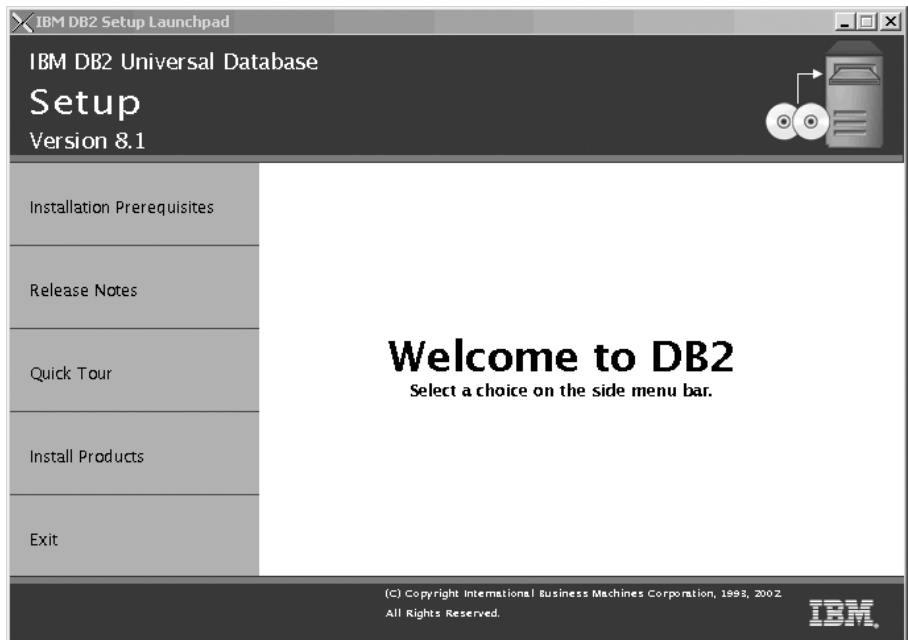
To install the DB2 online information using the DB2 Setup wizard:

1. Log on to the system as a user with root authority.
2. Change to the directory where the CD-ROM is mounted by entering the following command:

```
cd /cdrom
```

where */cdrom* represents mount point of the CD-ROM.

3. Enter the **./db2setup** command to start the DB2 Setup wizard. After a few moments, the IBM DB2 Setup Launchpad opens.



From this window, you can view installation prerequisites and the release notes, you can take a Quick Tour to explore the features of DB2 Universal Database Version 8, or you can proceed directly to the installation. You may want to review the installation prerequisites and release notes for late-breaking information.

Once you have initiated the installation, proceed through the DB2 Setup wizard installation panels and make your selections. Installation help is available to guide you through the remaining steps. To invoke the installation help, click **Help** or press F1. You can click **Cancel** at any time

to end the installation. DB2 files will only be copied to your system once you have clicked **Finish** on the last DB2 Setup wizard installation panel.

Related concepts:

- “Installation overview for DB2 servers (UNIX)” on page 52
- “Installation overview for a partitioned DB2 server (UNIX)” on page 64
- “Installation overview for DB2 Personal Edition (Linux)” in the *Quick Beginnings for DB2 Personal Edition*

Installing Multipartition Database Environments on the AIX Platform

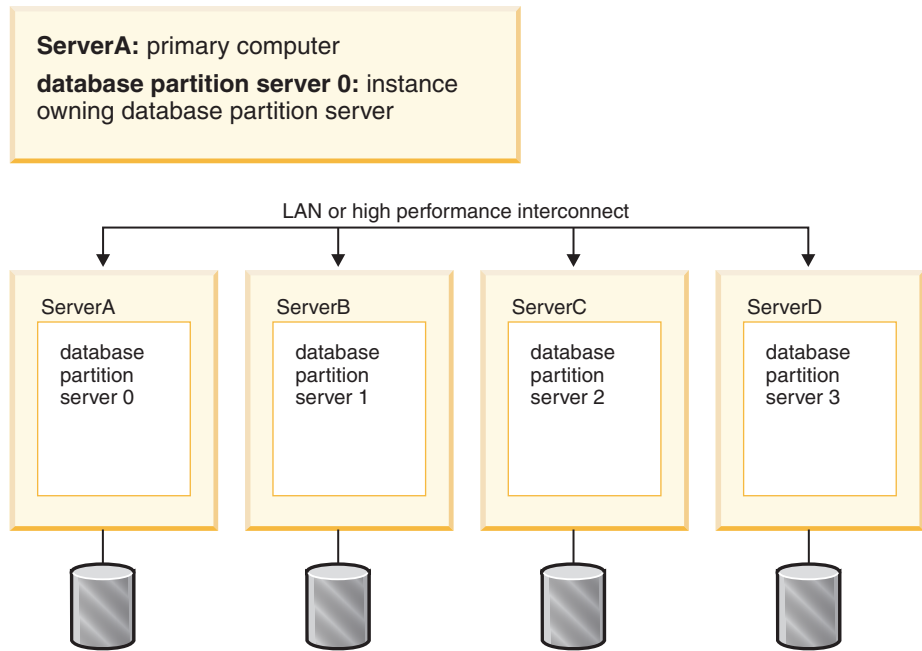
The sections that follow describe how to multipartition database environments on the AIX platform.

In a satellite environment, the DB2 Enterprise Server Edition product can function as the satellite control server when it is installed with the Satellite Control Server component on the AIX platform. The satellite control server can be a partitioned database environment.

Installation overview for a partitioned DB2 server (UNIX)

The following diagram shows a DB2® Enterprise Server Edition (ESE) configuration with four database partition servers, one per computer. Setup instructions are based on this configuration but can be easily adjusted for partitioned configurations with a fewer or greater number of computers and

database partition servers.



ServerA will be referred to as the primary or instance owning computer. ServerB, ServerC, and ServerD will be referred to as participating computers.

Installation overview:

Preparing your environment for installation

Before you install DB2, you must prepare your environment for the installation. In some work environments, the System Administrator will perform these tasks. To prepare your environment:

1. Verify that each computer meets the necessary operating system, memory, and disk requirements.
2. Update kernel parameters on each computer (HP-UX, Linux, and Solaris only).
3. Modify environment settings on each computer (AIX only).
4. Create a DB2 home file system on the primary computer (ServerA) and share it with participating computers (ServerB, ServerC, ServerD). This will be the instance home directory.
5. Create required users and groups on each computer.

Installing DB2

After preparing your environment, you will install DB2 on each computer. It is recommended that you install the instance owning partition first and create a response file which is used to install on the

other computers. This ensures that the same components are installed and configured the same way. However, the other participating computers can also be installed by using the DB2 Setup wizard, making the same component choices and not creating an instance.

It is recommended that you create a local administration contact list on the instance owning partition. When the DB2 Administration Server is installed and configured on the other participating computers, it will be configured to use the contact list on the instance owning computer.

To install DB2 on each computer using the recommended method, you will:

1. Mount the product CD-ROM on the primary computer (ServerA) and copy the contents to a directory on the shared DB2 home file system, where it can be accessed by the participating computers.
2. Install DB2 on the primary computer using the DB2 Setup wizard. The DB2 Setup wizard allows you to select features, create a DB2 instance, specify configuration settings, and create a response file for installing DB2 on participating computers.
3. Save the response file you created with the DB2 Setup wizard to a directory on your DB2 home file system, where it can be accessed by the participating computers.
4. Log on to each of the participating computers and perform a response file installation using the response file you created. The response file and DB2 product CD-ROM contents will be available to the participating computers on the shared DB2 home file system.

Post-installation setup

Once DB2 has been installed on all computers, there are a number of post-installation setup tasks. To setup DB2 after installation:

1. Update the node configuration file (`db2nodes.cfg`). When you install DB2 on the primary computer using the DB2 Setup wizard, you will create a DB2 instance. The information you provide in the node configuration file tells DB2 which database partition servers will participate in the instance.
2. Enable communications between the database partition servers. This requires that you update the `/etc/services` file on each computer.
3. Enable the execution of remote commands. This allows each database partition server to perform remote commands on the other database partition servers. This task must be performed on each computer.

Verifying the installation

After your system setup is complete, verifying the installation is recommended. To verify the installation:

1. Create a sample database.
2. Run SQL commands to retrieve information from the sample database and ensure that the sample data has been evenly distributed to all database partition servers.

Related concepts:

- “Instance creation” in the *Administration Guide: Implementation*
- “Installation overview for DB2 servers (UNIX)” on page 52

Related tasks:

- “Initializing a warehouse control database during installation” in the *Data Warehouse Center Administration Guide*
- “Tools catalog database and DAS scheduler setup and configuration” in the *Administration Guide: Implementation*
- “Notification and contact list setup and configuration.” in the *Administration Guide: Implementation*
- “Installing a partitioned DB2 server (AIX)” on page 67
- “Installing a partitioned DB2 server (HP-UX)” in the *Quick Beginnings for DB2 Servers*
- “Installing a partitioned DB2 server (Linux)” in the *Quick Beginnings for DB2 Servers*
- “Installing a partitioned DB2 server (Solaris)” in the *Quick Beginnings for DB2 Servers*

Related reference:

- “UPDATE HEALTH NOTIFICATION CONTACT LIST Command” in the *Command Reference*

Installing a partitioned DB2 server (AIX)

This topic outlines steps for installing a partitioned DB2 Enterprise Server Edition server on AIX.

Prerequisites:

Ensure that your computer meets the following requirements:

1. Installation requirements for partitioned DB2 servers
2. Memory requirements for partitioned DB2 servers
3. Disk requirements for partitioned DB2 servers

4. Groups and user accounts for DB2 installations

See the Related references for more information.

Procedure:

It is recommended that you read the Installation overview for partitioned DB2 servers prior to beginning the installation.

To install a partitioned DB2 server on AIX:

1. Update the AIX environment settings for a partitioned DB2 installation.
2. Verify that NFS is running.
3. Create a DB2 home file system for a partitioned database system.
4. Create required users for a partitioned DB2 installation.
5. Mount the DB2 CD-ROM.
6. *Optional:* Copy the contents of the DB2 product CD-ROM to your computer.
7. Install a database partition server on the primary computer using the DB2 Setup wizard.
8. Install database partition servers on participating computers using a response file.
9. Update the node configuration file (db2nodes.cfg).
10. Enable communication between database partition servers.
11. Enable the execution of remote commands.
12. Enable Control Center administration.
13. *Optional:* Apply the latest FixPak.
14. *Optional:* Verify the partitioned database installation.
15. *Optional:* Install the DB2 online documentation.

Related concepts:

- “Installation overview for a partitioned DB2 server (UNIX)” on page 64

Related tasks:

- “Updating AIX environment settings for a partitioned DB2 installation” on page 73
- “Verifying that NFS is running (AIX)” on page 75
- “Creating a DB2 home file system for a partitioned database system (AIX)” on page 76
- “Creating required users for a partitioned DB2 server installation (AIX)” on page 78
- “Mounting the DB2 CD-ROM (AIX)” on page 58

- “Copying the contents of the DB2 product CD-ROM to your computer” on page 80
- “Installing a database partition server on the primary computer using the DB2 Setup wizard (UNIX)” on page 81
- “Installing database partition servers on participating computers using a response file (UNIX)” on page 86
- “Updating the node configuration file (UNIX)” on page 87
- “Enabling communications between database partition servers” on page 89
- “Enabling the execution of remote commands (UNIX)” on page 90
- “Enabling Control Center administration (UNIX)” on page 91
- “Applying the latest FixPak” on page 25
- “Verifying a partitioned database server installation (UNIX)” on page 93
- “Installing DB2 online documentation (UNIX)” on page 62
- “Creating group and user IDs for a DB2 installation” in the *Installation and Configuration Supplement*
- “Setting up a working collective to distribute commands to ESE workstations (AIX)” in the *Quick Beginnings for DB2 Servers*

Related reference:

- “Disk requirements for a partitioned DB2 server (UNIX)” on page 72
- “Memory requirements for partitioned DB2 servers (UNIX)” on page 71
- “Installation requirements for partitioned DB2 servers (AIX)” on page 69

Installation requirements for partitioned DB2 servers (AIX)

This topic lists hardware, operating system, software and communication requirements for a partitioned DB2 server (AIX).

Hardware requirements

DB2 supports the following hardware:

- IBM RISC/6000
- eServer pSeries

Operating system requirements

DB2 Enterprise Server Edition is available on:

- AIX Version 4.3.3 with maintenance level 9 or later (32-bit)
- AIX Version 5.1.0 with maintenance level 2 or later (32-bit and 64-bit)

The following AIX filesets are required to install or run DB2 in languages other than English:

- X11.fnt.ucs.ttf (AIX Windows Unicode TrueType Fonts)

- x1C.rte 5.0.2.x
- For Asian languages, the following filesets are also required:
 - X11.fnt.ucs.ttf_CN (for zh_CN or Zh_CN)
 - X11.fnt.ucs.ttf_KR (for ko_KR)
 - X11.fnt.ucs.ttf_TW (for zh_TW or Zh_TW)
- On AIX Version 4.3.3 the following fileset is required:
 - x1C.aix43.rte 5.0.2.x
- On AIX Version 5L the following fileset is required:
 - x1C.aix50.rte 5.0.2.x

AIX filesets can be downloaded from:

<http://techsupport.services.ibm.com/server/fixes>

Software requirements

- Java Runtime Environment (JRE) Version 1.3.1 is required to run DB2 servers and DB2's Java-based tools, such as the Control Center.
- If you plan to use the Tivoli Storage Manager facilities back up and restore to your databases, you require the Tivoli Storage Manager Client Version 4.2.0 or later.
- A browser is required to view online help.

Communication requirements

You can use APPC, TCP/IP or MPTN (APPC over TCP/IP). To remotely administer a Version 8 DB2 database, you must connect using TCP/IP. DB2 Version 8 servers, using the DB2 Connect server support feature, support only outbound client APPC requests; there is no support for inbound client APPC requests.

- For TCP/IP connectivity, no additional software is required.
- For APPC (CPI-C) connectivity, through the DB2 Connect server support feature, one of the following communication products is required:
 - IBM eNetwork Communications Server for AIX V5.0.3
 - Bull DPX/20 SNA/20
- For LDAP (Lightweight Directory Access Protocol) support, you require an IBM SecureWay Directory Client V3.1.1
- If you plan to use the Simple Network Management Protocol (SNMP) subagent, you require DPI 2.0 provided by IBM SystemView Agent.

DB2 Administration Server (DAS) requirements

The following requirements must be met:

- A DAS must be created on each physical machine for the Control Center and the Task Center to work properly.

- Each DAS must be created under a userID (same as an instance).
- If the same userID is to be used on all physical machines, then that userID's home directory cannot be shared (cross mounted) with the other machines.
- If a different userID is used for each DAS, then the home directories of the userIDs that are used can be shared (cross mounted).
- As long as a DAS is created on each machine, it does not matter whether:
 - A different userID is used for each DAS, or
 - The same userID is used and that the userID's home directory is not shared.

Installing DB2 products or sharing instance directory on NFS

Currently, we do not support the installation of DB2 products on NFS. Installing DB2 on NFS (for example, NFS mounting /usr/opt/db2_08_01 or /opt/IBM/db2/V8.1) can be error prone and these errors can be difficult to diagnose.

The following configuration is not supported:

- Setting up an instance on a filesystem
- NFS mounting a filesystem from multiple machines, and then run DB2 on these machines using that same instance.

This configuration can cause file locking and performance problems.

Related tasks:

- “Installing a partitioned DB2 server (AIX)” on page 67

Memory requirements for partitioned DB2 servers (UNIX)

At a minimum, DB2 requires 256 MB of RAM. Additional memory may be required. In a partitioned database environment, the amount of memory required for each database partition server depends heavily on your configuration.

When determining memory requirements, be aware of the following:

- Additional memory may be required for non-DB2 software that may be running on your system.
- Additional memory is required to support database clients.
- Specific performance requirements may determine the amount of memory needed.
- Memory requirements will be affected by the size and complexity of your database system.
- Memory requirements will be affected by the extent of database activity and the number of clients accessing your system.

- Memory requirements in a partitioned environment may be affected by system design. Demand for memory on one computer may be greater than the demand on another.

Related tasks:

- “Installing a partitioned DB2 server (AIX)” on page 67
- “Installing a partitioned DB2 server (HP-UX)” in the *Quick Beginnings for DB2 Servers*
- “Installing a partitioned DB2 server (Linux)” in the *Quick Beginnings for DB2 Servers*
- “Installing a partitioned DB2 server (Solaris)” in the *Quick Beginnings for DB2 Servers*

Disk requirements for a partitioned DB2 server (UNIX)

Disk requirements vary depending on your file system and the type of installation you perform. The DB2 Setup wizard provides typical, data warehouse typical, satellite typical, compact, and custom installation types. The following table provides an approximate disk space requirement for each installation type.

Table 6. Disk requirements for a partitioned DB2 server

Installation type	required disk space
Typical	450 to 500MB
Compact	300 to 350 MB
Custom	200 MB to 800 MB

Typical installation

DB2 is installed with most features and functionality, using a typical configuration. Includes graphical tools such as the Control Center and Configuration Assistant.

Compact installation

Only the basic DB2 features and functions are installed. Does not include graphical tools.

Custom installation

A custom installation allows you to select the features you want to install.

Remember to include disk space allowance for required software, communication products, and documentation. For DB2 version 8, documentation is provided on separate CD-ROMS.

Related tasks:

- “Installing a partitioned DB2 server (AIX)” on page 67
- “Installing a partitioned DB2 server (HP-UX)” in the *Quick Beginnings for DB2 Servers*
- “Installing a partitioned DB2 server (Linux)” in the *Quick Beginnings for DB2 Servers*
- “Installing a partitioned DB2 server (Solaris)” in the *Quick Beginnings for DB2 Servers*

Updating AIX environment settings for a partitioned DB2 installation

This task describes the environment settings that you need to update on each computer that will participate in your partitioned database system.

Procedure:

To update AIX environment settings:

1. Log on to the computer as a user with root authority.
2. Set the AIX maxuproc (maximum number of processes per user) device attribute to 4096 by entering the following command:

```
chdev -l sys0 -a maxuproc='4096'
```

3. Set the TCP/IP network parameters on all the workstations that are participating in your partitioned database system to the following values:

```
thewall      = 65536
sb_max       = 1310720
rfc1323      = 1
tcp_sendspace = 221184
tcp_recvspace = 221184
udp_sendspace = 65536
udp_recvspace = 65536
ipqmaxlen    = 250
somaxconn    = 1024
```

To list the current settings of all network-related parameters, enter the following command:

```
no -a | more
```

To set a parameter, enter the follow command:

```
no -o parameter_name=value
```

where:

- *parameter_name* represents the parameter you want to set.
- *value* represents the value that you want to set for this parameter.

For example, to set the tcp_sendspace parameter to 221184, enter the following command:

```
no -o tcp_sendspace=221184
```

The above values are the minimum values for these parameters. If any of the network-related parameters are already set to a higher value, do not change it.

4. If you are using a high speed interconnect, you must set the *spoolsize* and *rpoolsz* for *css0* to the following values:

```
spoolsize    16777216
rpoolsz      16777216
```

To list the current settings of these parameters, enter the following command:

```
lsattr -l css0 -E
```

To set these parameters, enter the following commands:

```
/usr/lpp/ssp/css/chgcss -l css0 -a spoolsize=16777216
/usr/lpp/ssp/css/chgcss -l css0 -a rpoolsz=16777216
```

If you are not using the */tftpboot/tuning.cst* file to tune your system, you can use the */opt/lpp/db2_08_01/misc/rc.local.sample* script file to update the network-related parameters after installation. To update the network-related parameters using the sample script file after installation, perform the following steps:

- a. Copy this script file to the */etc* directory and make it executable by root by entering the following commands:

```
cp /opt/lpp/db2_08_01/misc/rc.local.sample /etc/rc.local
chown root:sys /etc/rc.local
chmod 744 /etc/rc.local
```
 - b. Review the */etc/rc.local* file and update it if necessary.
 - c. Add an entry to the */etc/inittab* file so that the */etc/rc.local* script is executed whenever the machine is rebooted. You can use the **mkinitrd** command to add an entry to the */etc/inittab* file. To add this entry, enter the following command:

```
mkinitrd "rclocal:2:wait:/etc/rc.local > /dev/console 2>&1"
```
 - d. Ensure that */etc/rc.nfs* entry is included in the */etc/inittab* file by entering the following command:

```
lsinitrd rcnfs
```
 - e. Update the network parameters without rebooting your system by entering the following command:

```
/etc/rc.local
```
5. Ensure that you have enough paging space for a partitioned installation of DB2 ESE to run. If you do not have sufficient paging space, the operating

system will kill the process that is using the most virtual memory (this is likely to be one of the DB2 processes). To check for available paging space, enter the following command:

```
lspv -a
```

This command will return output similar to the following:

Page Space	Physical Volume	Volume Group	Size	%Used	Active	Auto	Type
paging00	hdisk1	rootvg	60MB	19	yes	yes	lv
hd6	hdisk0	rootvg	60MB	21	yes	yes	lv
hd6	hdisk2	rootvg	64MB	21	yes	yes	lv

We recommend that the paging space available be equal to twice the amount of physical memory installed on your computer.

6. If you are creating a small to intermediate size partitioned database system, the number of network file system daemons (NFSDs) on the instance-owning computer should be close to:

of biod on a computer * # of computers in the instance

We recommended that you run 10 biod processes on every computer. According to the above formula, on a four computer system with 10 biod processes, you would use 40 NFSDs.

If you are installing a larger system, you can have up to 120 NFSDs on the computer.

For additional information about NFS, refer to your NFS documentation.

Verifying that NFS is running (AIX)

Network File System (NFS) must be running on each computer.

Procedure:

To verify that Network File System (NFS) is running on each computer that will participate in your partitioned database system, enter the following command on each computer:

```
lssrc -g nfs
```

The Status field for NFS processes should indicate active. More specifically, DB2 requires that the following two NFS processes are active:

```
rpc.lockd  
rpc.statd
```

If these processes are not running, consult your AIX operating system documentation.

Creating a DB2 home file system for a partitioned database system (AIX)

This task describes how to create a DB2 home file system, NFS export the home file system, and NFS mount the home file system from each participating computer.

It is recommended that you create a home file system that is 1 GB in size or greater. Later installation instruction will ask that you copy the contents of the DB2 product CD-ROM to a directory on your DB2 home file system. The DB2 product CD-ROM will temporarily occupy approximately 700 MB of space. A DB2 instance will require at least 50 MB of space. If you do not have 1 GB of free space, you can mount the DB2 product CD-ROM from each participating computer as an alternative to copying the contents to disk.

Prerequisites:

You must have:

- root authority to create a file system
- Created a volume group where your file system is to physically reside.

Procedure:

To create, NFS export, and NFS mount the DB2 home file system, perform the following steps:

Creating the DB2 home file system

Log on to the primary computer (ServerA) in your partitioned database system as a user with root authority and create a home file system for your partitioned database system called /db2home.

1. Enter the **smit jfs** command.
2. Click on the **Add a Journaled File System** icon.
3. Click on the **Add a Standard Journaled File System** icon.
4. Select an existing volume group from the **Volume Group Name** list where you want this file system to physically reside.
5. Set the **SIZE of file system (in 512-byte blocks) (Num.)** field to 180 000 (this is about 90 MB).
6. Enter the mount point for this file system in the **MOUNT POINT** field. In this example, the mount point is /db2home.
7. Set the **Mount AUTOMATICALLY at system restart** field to yes.
The remaining fields can be left to the default settings.
8. Click **OK**.

Exporting the DB2 home file system

1. NFS export the /db2home file system so that it is available to all of the computers that will participate in your partitioned database system:
 - a. Enter the **smit nfs** command.
 - b. Click on the **Network File System (NFS)** icon.
 - c. Click on the **Add a Directory to Exports List** icon.
 - d. Enter the pathname and directory to export (for example, /db2home) in the **PATHNAME of directory to export** field.
 - e. Enter the name of each workstation that will participate in your partitioned database system in the **HOSTS allowed root access** field. Use a comma (,) as the delimiter between each name. For example, ServerA, ServerB, ServerC. If you are using a high speed interconnect, we recommend that you specify the high speed interconnect names for each workstation in this field as well. The remaining fields can be left to the default settings.
 - f. Click **OK**.
2. Log out.

Mounting the DB2 home file system from each participating computer

Log on to *each* participating computer (ServerB, ServerC, ServerD) and NFS mount the file system that you exported by performing the following steps:

1. Enter the **smit nfs** command.
2. Click on the **Network File System (NFS)** icon.
3. Click on the **Add a File System for Mounting** icon.
4. Enter the pathname of the mount point in the **PATHNAME of the mount point (Path)** field.

The path name of the mount point is where you should create the DB2 home directory. For this example, use/db2home.
5. Enter the pathname of the remote directory in the **PATHNAME of the remote directory** field.

For our example, you should enter the same value that you entered in the **PATHNAME of the mount point (Path)** field.
6. Enter the *hostname* of the machine where you exported the file system in the **HOST where the remote directory resides** field.

This is the hostname of the machine where the file system that you are mounting was created.

To improve performance, you may want to NFS mount the file system that you created over a high speed interconnect. If you

want to mount this file system using a high speed interconnect, you must enter its name in the **HOST where remote directory resides** field.

You should be aware that if the high speed interconnect ever becomes unavailable for some reason, every workstation that participates in your partitioned database system will lose access to these DB2 home directory.

7. Set the **MOUNT now, add entry to /etc/filesystems or both?** field to both.
8. Set the **/etc/filesystems entry will mount the directory on system RESTART** field to yes.
9. Set the **MODE for this NFS file system** field to read-write.
10. Set the **Mount file system soft or hard** field to soft.
A soft mount means that the computer *will not* try for an infinite period of time to remotely mount the directory. A hard mount means that your machine will infinitely try to mount the directory. This could cause problems in the event of a system crash. We recommend that you set this field to soft.
The remaining fields can be left to the default settings.
11. Ensure that this file system is mounted with the **Allow execution of SUID and sgid programs in this file system?** field set to Yes. This is the default setting.
12. Click **OK**.
13. Log out.

Related tasks:

- “Copying the contents of the DB2 product CD-ROM to your computer” on page 80

Creating required users for a partitioned DB2 server installation (AIX)

This task is part of the larger task of *Installing a partitioned DB2 server on AIX*.

Three users and groups are required to operate DB2. The user and group names used in the following instructions are documented in the following table. You may specify your own user and group names as long as they adhere to your system naming rules and DB2 naming rules.

Table 7. Required users and groups

Required user	user name	group name
Instance owner	db2inst1	db2iadm1
Fenced user	db2fenc1	db2fadm1

Table 7. Required users and groups (continued)

Required user	user name	group name
Administration server user	db2as	db2asgrp

If an existing user is used as the Administration server user, this user must also exist on the all participating computers before installation. If you use the DB2 Setup wizard to create a new user for the Administration server on the instance owning computer, then this user will also be created (if necessary) during the response file installations on the participating computers. If the user already exists on the participating computers, it must have the same primary group.

Prerequisites:

- You must root authority to create users and groups.
- If you manage users and groups with NIS/NIS+ or similar security software, see *NIS/NIS+ considerations* before creating users and groups. Additional steps may be required to when defining DB2 users and groups.

Restrictions:

The user names you create must conform to both your operating system’s naming rules, and those of DB2.

Procedure:

To create all three of these users, perform the following steps:

1. Log on to the primary computer.
2. Create a group for the instance owner (for example, db2iadm1), the user that will execute UDFs or stored procedures (for example, db2fadm1), and the Administration Server (for example, db2asgrp) by entering the following commands:


```

mkgroup id=999 db2iadm1
mkgroup id=998 db2fadm1
mkgroup id=997 db2asgrp
            
```
3. Create a user that belongs to each group that you created in the previous step using the following commands. The home directory for each user will be the DB2 home directory that you previously created and shared (db2home).

```

mkuser id=1004 pgrp=db2iadm1 groups=db2iadm1 home=/db2home/db2inst1
  core=-1 data=491519 stack=32767 rss=-1 fsize=-1 db2inst1
mkuser id=1003 pgrp=db2fadm1 groups=db2fadm1 home=/db2home/db2fenc1
  db2fenc1
mkuser id=1002 pgrp=db2asgrp groups=db2asgrp home=/db2home/db2as
  db2as
            
```

4. Set an initial password for each user that you created by entering the following commands:

```
passwd db2inst1
passwd db2fenc1
passwd db2as
```
5. Log out.
6. Log on to the primary computer as each user that you created (db2inst1, db2fenc1, and db2as). You may be prompted to change each user's password since this is the first time that these users have logged onto the system.
7. Log out.
8. Create the exact same user and group accounts on each computer that will participate in your partition database system. For our example, perform this task on ComputerB, ComputerC, and ComputerD.

Related reference:

- "NIS installation considerations" in the *Quick Beginnings for DB2 Servers*

Mounting the DB2 CD-ROM (AIX)

You must mount the DB2 product CD-ROM before you can start the DB2 Setup wizard.

Procedure:

To mount the DB2 installation CD and copy the contents:

1. Create a directory for the CD-ROM by entering the following command:

```
mkdir /cdrom -p
```

2. Allocate a CD-ROM file system by entering the following command:

```
crfs -v cdrfs -p ro -d cd0 -m /cdrom
```

where cd0 is the standard representation for the CD-ROM drive.

3. Mount the CD-ROM file system by entering the following command:

```
mount /cdrom
```

Copying the contents of the DB2 product CD-ROM to your computer

This task describes the steps for copying the contents of the DB2 ESE product CD-ROM to the shared DB2 home file system. Copying the contents of the DB2 CD-ROM is a step unique to partitioned installations of DB2. Because you are likely to be installing DB2 to multiple computers simultaneously, installing from hard disk is significantly faster than installing from a CD-ROM. This method is recommended for any system that includes more than four computers.

The alternative is to NFS mount the CD-ROM file system from each computer. You may want to mount the CD-ROM from each computer if you do not have enough disk space on the DB2 home file system or if you are installing on fewer than four computers.

Procedure:

To mount the DB2 installation CD and copy the contents

1. Create a directory on your /db2home file system for the DB2 product CD-ROM:

```
mkdir /db2home/db2cdrom
```

2. Copy the contents of the CD-ROM to directory that you created:

```
cp -R /cdrom /db2home/db2cdrom
```

Installing a database partition server on the primary computer using the DB2 Setup wizard (UNIX)

This task describes how to launch the DB2 Setup wizard and install a DB2 ESE database partition server on the primary computer in your partitioned system. Information is provided for specific DB2 Setup wizard panels that are key to setting up your partitioned database system. Not all of the DB2 Setup wizard panels are documented in this topic. Use the DB2 Setup wizard installation help when in doubt.

Prerequisites:

You must have root authority to install DB2.

Refer to the CD-ROM label to ensure that you are using the CD-ROM with your appropriate language.

During instance creation a number of ports equal to the number of logical nodes that the instance is capable of supporting will be reserved in the /etc/services. These ports will be used by the Fast Communication Manager. The reserved ports will be in the following format:

```
DB2_InstanceName  
DB2_InstanceName_1  
DB2_InstanceName_2  
DB2_InstanceName_END
```

The only mandatory entries are the beginning (DB2_InstanceName) and ending (DB2_InstanceName_END) ports. The other entries are reserved in the services file so that other applications do not use these ports.

Procedure:

To install DB2 ESE on the primary computer using the DB2 Setup wizard:

1. On AIX, HP-UX and Linux, from the directory on the /db2home file system where you copied the contents of the DB2 product CD-ROM, enter the **db2setup** command to start the DB2 Setup wizard. On the Solaris Operating Environment, from the directory on the /db2home file system where you copied the contents of the DB2 product CD-ROM, enter the following command to start the DB2 Setup wizard:

```
zcat product.tar.Z | tar -xf - ; ./product/db2setup
```

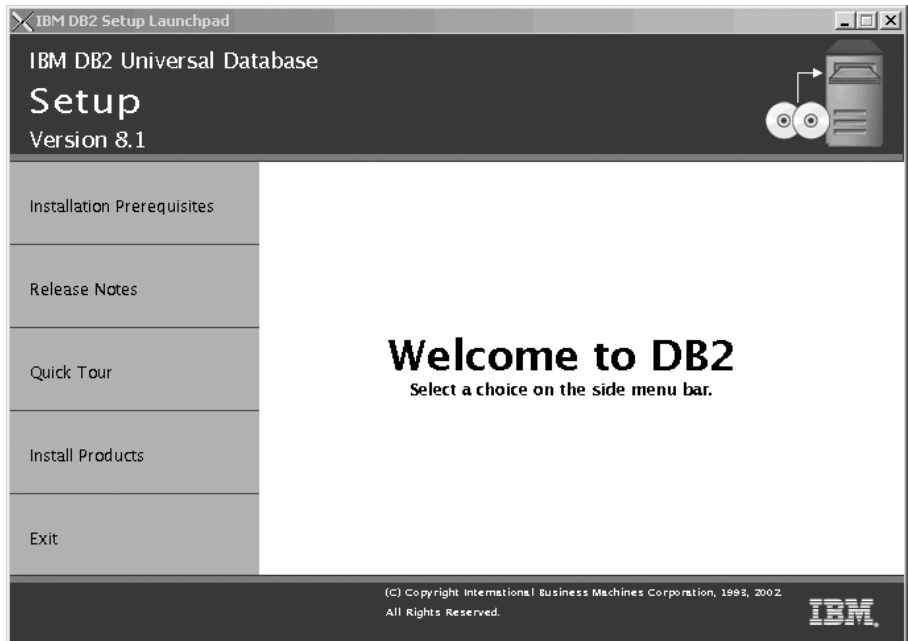
For example, if the product name for DB2 Enterprise Server Edition is *ese*, then enter the following command:

```
zcat ese.tar.Z | tar -xf - ; ./ese/db2setup
```

For example:

```
/db2home/db2cdrom/db2setup
```

After a few moments, the DB2 Version 8 Installation Launchpad opens.



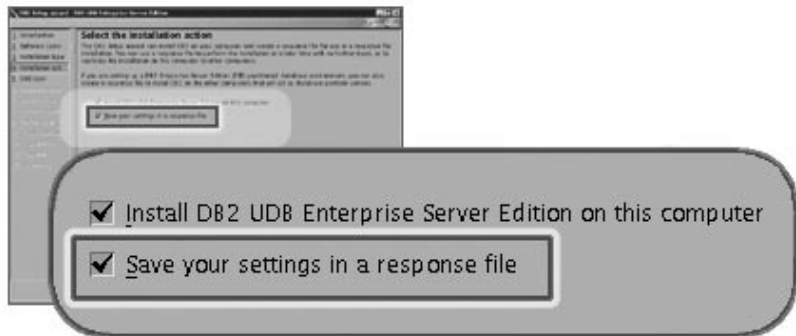
From the DB2 Launchpad, you can view the Installation Prerequisites and Release Notes. You can also take a Quick Tour to learn about DB2 Version 8 features.

2. When you have finished viewing Launchpad information, proceed with the installation.

The following list provides information about specific DB2 Setup wizard installation panels and the selections you must make to correctly install DB2 ESE on your primary computer.

Select the installation action

On the *Select the installation action* panel, you must select both **Install DB2 UDB Enterprise Server Edition on this computer** and **Save your setting to a response file**. The response file will be used to install DB2 on participating computers.



Set user information for the DB2 Administration Server (DAS)

On the *Set user information for the DB2 Administration Server (DAS)* panel you must select the DAS user that you created when you prepared your environment for installation. To do this, select the **Existing user** radio button and enter the user or use the ... button to locate the DAS user you previously created.



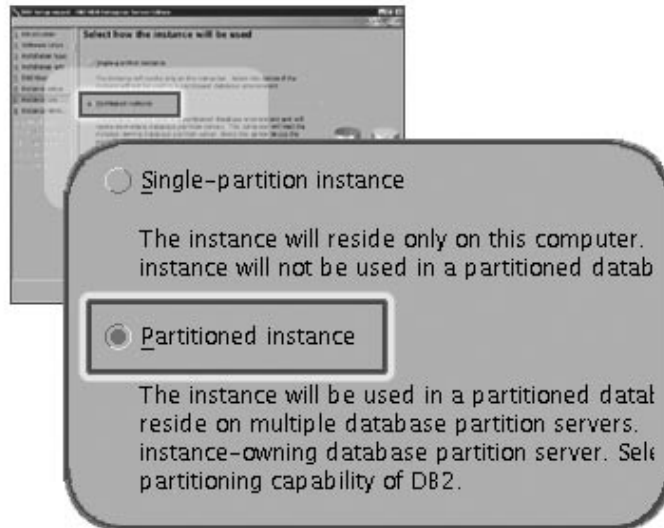
Set up a DB2 instance

On the *Set up a DB2 instance* panel, select **Create a DB2 instance**.

Select how this instance will be used

On the *Select how this instance will be used* panel, you must select

Partitioned instance.



Set user information for the DB2 instance

On the *Set user information for the DB2 instance* panel, you must select the instance owner that you created when you prepared your environment for installation. To do this, select the **Existing user** radio button and enter the user or use the ... button to select the instance owner.



Set user information for the fenced user

On the *Set user information for the fenced user* panel, select the existing fenced user that you created when you prepared your environment for installation. Select the Existing user radio button and enter the

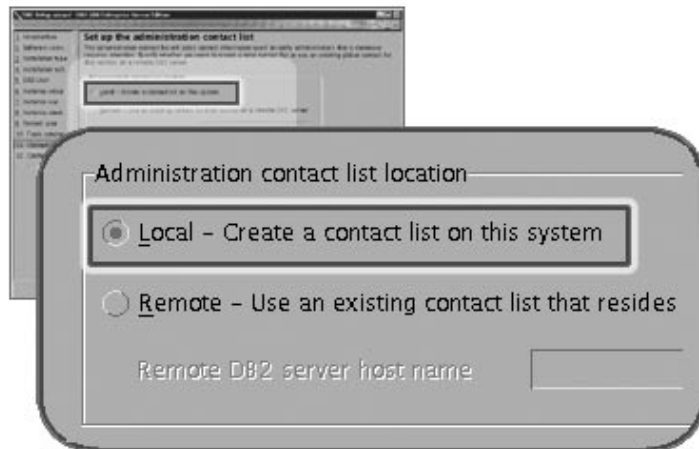
user or use the ... button to select the fenced user.



Set up an administration contact list panel

On the *Set up an administration contact list panel*, select **Local**. This selection will create a file on the primary computer that will store contact information for your system.

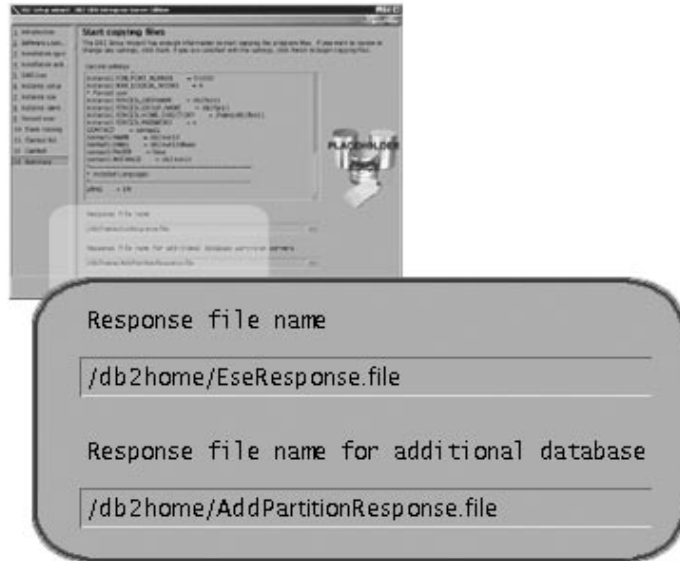
The contact information is used by DB2 to send notifications and alerts to a system administrator. You can specify notification and alert parameters after setup is complete. Participating computers will remotely access this contact list on the primary computer.



Start copying files

On the *Start copying files panel* you must specify a location and name for two response files. The first response file is for installing a replica of the primary computer installation. The second response file is for installing database partition servers on participating computers. You

can place the first response file where you like. The second response file, which we have named `AddPartitionResponse.file`, must be saved to the `/db2home` directory, where it will be accessible to participating computers.



The next step in installing an ESE partitioned database system is to use the response file you created (`AddPartitionResponse.file`) to install database partition servers on the participating computers.

Related reference:

- “Supported DB2 interface languages, locales, and code pages” in the *Quick Beginnings for DB2 Servers*
- “db2setup - Install DB2 Command” in the *Command Reference*

Installing database partition servers on participating computers using a response file (UNIX)

This task is part of the larger task of *Installing a DB2 ESE partitioned server*.

In this task you will use the response file you created using the DB2 Setup wizard to install database partition servers on participating computers.

Prerequisites:

- You have installed DB2 on the primary computer using the DB2 Setup wizard and have created a response file for installing on participating computers.

- You must have root authority on participating computers.

Procedure:

To install additional database partition servers using a response file:

1. As root, log on to a computer that will participate in the partitioned database environment.
2. Change to the directory where you copied the contents of the DB2 product CD-ROM:

```
cd /db2home/db2cdrom
```

3. Enter the **./db2setup** command as follows:

```
./db2setup -r /responsefile_directory/response_file_name
```

In our example, we saved the response file, *AddPartitionResponse.file*, to the */db2home* directory. The command for our example, would be:

```
./db2setup -r /db2home/AddPartitionResponse.file
```

4. Check the messages in the log file when the installation finishes. The location of the log file is: */tmp/db2setup.log*

You must log onto each participating computer and perform a response file installation.

Related tasks:

- “Installing a database partition server on the primary computer using the DB2 Setup wizard (UNIX)” on page 81

Updating the node configuration file (UNIX)

The node configuration file (*db2nodes.cfg*), located in the instance owner’s home directory contains configuration information that tells DB2 which database partition servers participate in an instance. There is a *db2nodes.cfg* file for each instance in a partitioned database environment.

The *db2nodes.cfg* file must contain one entry for each database partition server that will participate in the instance. When you create an instance, the *db2nodes.cfg* file is automatically created and an entry for the instance-owning database partition server is added.

For example, when you created the DB2 instance using the DB2 Setup wizard, on the primary computer *ServerA*, the *db2nodes.cfg* file was updated as follows:

```
0      ServerA      0
```

This task provides steps for updating the *db2nodes.cfg* file to include entries for participating computers.

Prerequisites:

- DB2 must be installed on all participating computers.
- A DB2 instance must exist on the primary computer.
- You must be a user with SYSADM authority.
- If you plan to use a high speed switch for communication between database partition servers or if your partitioned configuration will have logical database partition servers, review the *DB2 node configuration file* topic for configuration examples and information about file format of `db2nodes.cfg`.

Procedure:

To update the `db2nodes.cfg` file:

1. Log on as the instance owner (in our example, `db2inst1` is the instance owner).
2. Ensure that the DB2 instance is stopped by entering:

```
INSTHOME/sql1lib/adm/db2stop
```

command, where *INSTHOME* is the home directory of the instance owner (the `db2nodes.cfg` file is locked when the instance is running and can only be edited when the instance is stopped).

For example, if your instance home directory is `/db2home/db2inst1`, enter the following command:

```
/db2home/db2inst1/sql1lib/adm/db2stop
```

3. Add an entry to the `db2nodes.cfg` file for each database partition server. When you first view the `db2nodes.cfg` file, it should contain an entry, similar to the following:

```
0      ServerA      0
```

This entry includes the database partition server number (node number), the TCP/IP host name of the server where the database partition server resides, and a logical port number for the database server partition.

If you are installing the partitioned configuration described in the installation overview, with four computers and a database partition server on each computer, the updated `db2nodes.cfg` should appear similar to the following:

```
0      ServerA      0
1      ServerB      0
2      ServerC      0
3      ServerD      0
```

4. When you have finished updating the `db2nodes.cfg` file, enter the `INSTHOME/sql1lib/adm/db2start` command, where *INSTHOME* is the

home directory of the instance owner. For example, if your instance home directory is `/db2home/db2inst1`, enter the following command:

```
/db2home/db2inst1/sqllib/adm/db2start
```

5. Log out.

Related reference:

- “DB2 node configuration file (db2nodes.cfg)” in the *Quick Beginnings for DB2 Servers*

Enabling communications between database partition servers

This task describes how to enable communication between the database partition servers that participate in your partitioned database system. Communication between database partition servers is handled by the Fast Communications Manager (FCM). To enable FCM, a port or port range must be reserved in the `/etc/services` file on each computer in your partitioned database system.

You must perform this task on participating computers only. When you create an instance using the DB2 Setup wizard, a port range is automatically reserved on the primary (instance-owning) computer.

Procedure:

1. Log on to the primary computer (instance owning computer) as a user with root authority.
2. View the default port range that has been reserved in the `/etc/services` file. It should appear similar to the following:

```
DB2_db2inst1      60000/tcp
DB2_db2inst1_1   60001/tcp
DB2_db2inst1_2   60002/tcp
DB2_db2inst1_END 60003/tcp.
```

By default, the first available four ports above 60000 are reserved. One for the instance-owning database partition server and three for logical database partition servers that you might choose to add to the computer after installation is complete.

DB2 port entries use the following format:

```
DB2_instance_name    port_number
```

where:

- *instance_name* is the name of the partitioned instance.
- *port_number* is the port number that you reserve for database partition server communications.

3. In turn, log onto each participating computer as a root user and add identical entries to the `/etc/services` file.

You can add a comment to describe each entry using the `#` comment identifier. For example:

```
DB2_db2inst1      60000/tcp  # instance-owning partition port
DB2_db2inst1_1   60001/tcp  # logical partition port
DB2_db2inst1_2   60002/tcp  # logical partition port
DB2_db2inst1_END 60003/tcp  # logical partition port
```

Related concepts:

- “Fast Communications Manager (UNIX)” in the *Quick Beginnings for DB2 Servers*

Related reference:

- “DB2 node configuration file (db2nodes.cfg)” in the *Quick Beginnings for DB2 Servers*

Enabling the execution of remote commands (UNIX)

In a partitioned database system, each database partition server must have the authority to perform remote commands on all the other database partition servers participating in an instance. This can be done by updating the `.rhosts` file in the home directory for the instance. Because the home directory for the instance is on the shared DB2 home file system, only one `.rhosts` file is required.

Prerequisites:

- You must have root authority.
- You must know the hostname of each participating computer
- You must know the instance owner’s user name.

Procedure:

1. Log onto the primary computer as a user with root authority.
2. Create a `.rhosts` file in the instance home directory. For example, if your instance home directory is `/db2home/db2inst1`, you can use an editor such as `vi` to create the `.rhosts` file by entering the following command:

```
vi /db2home/db2inst1/.rhosts
```

3. Add entries to the `.rhosts` file for each computer including the primary computer. The `.rhosts` file has the following format:

```
hostname  instance_owner_user_name
```


Some systems may require a long host name to be specified, for example: ServerA.yourdomain.com. Before you add hostname entries to the .rhosts file, make sure the hostnames in the /etc/hosts and the /etc/resolv.conf files can be resolved.

The *INSTHOME*/.rhosts file, should contain entries similar to the following:

```
ServerA.yourdomain.com db2inst1
ServerB.yourdomain.com db2inst1
ServerC.yourdomain.com db2inst1
ServerD.yourdomain.com db2inst1
```

Rather than specify each hostname individually, you may specify the following entry in the .rhosts file, but this may pose a security risk and should only be done in a test environment.

```
+ db2inst1
```

If you have specified a high speed switch (netname) in the db2nodes.cfg file, you should also add netname entries for each computer to the .rhosts file. The netname values are specified in the fourth column of the db2nodes.cfg file. A .rhosts file with high speed switch (netname) entries may look similar to the following:

```
ServerA.yourdomain.com db2inst1
ServerB.yourdomain.com db2inst1
ServerC.yourdomain.com db2inst1
ServerD.yourdomain.com db2inst1
Switch1.yourdomain.com db2inst1
Switch2.yourdomain.com db2inst1
Switch3.yourdomain.com db2inst1
Switch4.yourdomain.com db2inst1
```

An alternative to using a .rhosts file is to use /etc/hosts.equiv file. The /etc/hosts.equiv file would contain the exact same entries as the .rhosts file, but must be created on each computer.

For more information about the .rhosts file or the /etc/hosts.equiv file, see your operating system documentation.

Enabling Control Center administration (UNIX)

Before you can use the Control Center to administer your partitioned database system, you must start the DB2 Administration server on all computers.

Procedure: To enable Control Center administration for a partitioned database system:

Start the DB2 Administration Server on each computer

1. In turn, log on to each computer (ServerA, ServerB, ServerC, ServerD) as the DB2 Administration Server user. In our example, db2as is the DAS user.
2. Enter the following command to start the DB2 Administration Server:

```
/DASHOME/das/bin/db2admin start
```

where *DASHOME* is the home directory for the DB2 Administration Server. In our example, the *DASHOME* is */db2home/db2as*.

Applying the latest FixPak

Applying the latest FixPak is optionally part of the larger task of installing DB2 products.

A DB2 FixPak contains updates and fixes for bugs (Authorized Program Analysis Reports, or "APARs") found during testing at IBM, as well as fixes for bugs reported by customers. Every FixPak is accompanied by a document, called APARLIST.TXT, that describes the bug fixes it contains.

FixPaks are cumulative. This means that the latest FixPak for any given version of DB2 contains all of the updates from previous FixPaks for the same version of DB2. We recommend that you keep your DB2 environment running at the latest FixPak level to ensure problem-free operation.

When installing a FixPak on a partitioned ESE system, all participating computers must have the same FixPak installed while the system is offline.

Prerequisites:

Each FixPak may have specific prerequisites. See the FixPak README that accompanies the FixPak for more information.

Procedure:

1. Download the latest DB2 FixPak from the IBM DB2 UDB and DB2 Connect Online Support Web site at <http://www.ibm.com/software/data/db2/udb/winso2unix/support>.
2. Each FixPak contains a set of Release Notes and a README. The README provides instructions for installing the FixPak.

Verifying a partitioned database server installation (UNIX)

To verify that your DB2 server installation was successful, you will create a sample database and run SQL commands to retrieve sample data and to verify that the data has been distributed to all participating database partition servers.

Prerequisites:

You have completed all of the installation steps.

Procedure:

To create the SAMPLE database:

1. Log on to the primary computer (ServerA). as the instance-owning user. In our installation example, db2inst1 is the instance-owning user.
2. Enter the **db2sampl** command to create the SAMPLE database. By default, the sample database will be created in the instance-owner's home directory. In our example /db2home/db2inst1/ is the instance owner's home directory. The instance owner's home directory is the default database path.

This command may take a few minutes to process. There is no completion message; when the command prompt returns, the process is complete.

The SAMPLE database is automatically cataloged with the database alias SAMPLE when it is created.

3. Start the database manager by entering the **db2start** command.
4. Enter the following DB2 commands from a DB2 command window to connect to the SAMPLE database, retrieve a list of all the employees that work in department 20:

```
db2 connect to sample
db2 "select * from staff where dept = 20"
```

5. To verify that data has been distributed across database partition servers, enter the following commands from a DB2 command window:
select distinct dbpartitionnum(empno) from employee;

The output will list the database partitions used by the employee table. The specific output will depend on the number of partitions in the database and the number of partitions in the partition group that is used by the tablespace where the employee table was created.

After you have verified the installation, you can remove the SAMPLE database to free up disk space. Enter the **db2 drop database sample** command to drop the SAMPLE database.

Installing DB2 online documentation (UNIX)

This task describes how to install the DB2 online documentation using the DB2 Setup wizard on UNIX. The DB2 online documentation is installed separately from other DB2 products from its own CD-ROM.

Prerequisites:

Before you start the DB2 Setup wizard

- You require root authority to perform the installation.
- The DB2 product CD-ROM must be mounted on your system.
- The DB2 Setup wizard is a graphical installer. In order for it to run on your machine, you must have Xwindow software capable of rendering a graphical user interface.
- A Java Runtime Environment (JRE) must already be installed.

Procedure:

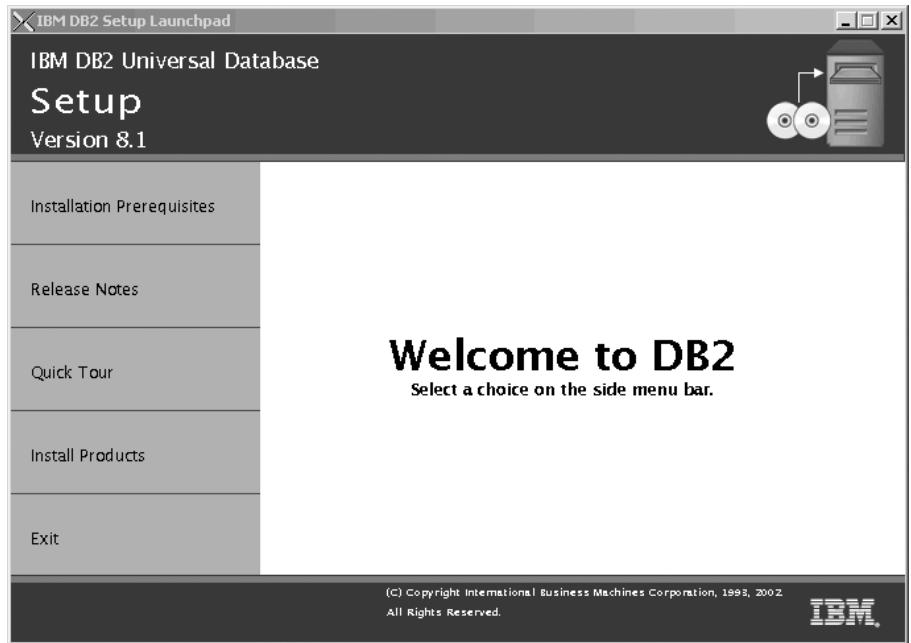
To install the DB2 online information using the DB2 Setup wizard:

1. Log on to the system as a user with root authority.
2. Change to the directory where the CD-ROM is mounted by entering the following command:

```
cd /cdrom
```

where */cdrom* represents mount point of the CD-ROM.

3. Enter the `./db2setup` command to start the DB2 Setup wizard. After a few moments, the IBM DB2 Setup Launchpad opens.



From this window, you can view installation prerequisites and the release notes, you can take a Quick Tour to explore the features of DB2 Universal Database Version 8, or you can proceed directly to the installation. You may want to review the installation prerequisites and release notes for late-breaking information.

Once you have initiated the installation, proceed through the DB2 Setup wizard installation panels and make your selections. Installation help is available to guide you through the remaining steps. To invoke the installation help, click **Help** or press F1. You can click **Cancel** at any time to end the installation. DB2 files will only be copied to you system once you have clicked **Finish** on the last DB2 Setup wizard installation panel.

Related concepts:

- “Installation overview for DB2 servers (UNIX)” on page 52
- “Installation overview for a partitioned DB2 server (UNIX)” on page 64
- “Installation overview for DB2 Personal Edition (Linux)” in the *Quick Beginnings for DB2 Personal Edition*

Chapter 3. Installing DB2 Personal Edition

Installing DB2 Personal Edition (Windows)	97	User accounts for installation and setup of DB2 Personal Edition	104
DB2 Personal Edition installation overview (Windows)	98	Starting the DB2 Setup wizard (Windows)	105
Installation requirements for DB2 Personal Edition (Windows)	100	Applying the latest FixPak	108
Memory requirements for DB2 Personal Edition (Windows)	102	Verifying the installation using the command line processor (CLP)	108
Disk requirements for DB2 Personal Edition (Windows)	102	Installing DB2 online documentation (Windows)	109
Extending the directory schema (Windows 2000 and Windows .NET)	103		

DB2 Personal Edition can function as a satellite when it is installed with the Satellite Synchronization Component.

Installing DB2 Personal Edition (Windows)

This topic outlines steps for installing DB2 Personal Edition on Windows.

Prerequisites:

Ensure that your computer meets the following requirements:

1. Installation requirements for DB2 Personal Edition (Windows)
2. Memory requirements for DB2 Personal Edition (Windows)
3. Disk requirements for DB2 Personal Edition (Windows)
4. User accounts for installation and setup of DB2 Personal Edition (Windows)

See the Related references at the bottom for more information.

Procedure:

It is recommended that you read the DB2 Personal Edition installation overview (Windows) prior to beginning the installation.

To install DB2 Personal Edition on Windows:

1. If you are installing on Windows 2000 or Windows .NET and intend to use Lightweight Directory Access Protocol (LDAP), you must extend the directory schema.
2. Install DB2 Personal Edition using the DB2 Setup Wizard (Windows).

3. *Optional*: Install the latest FixPak.
4. *Optional*: Verify the installation using the Command Line Processor (CLP).
5. *Optional*: Install the DB2 documentation (Windows).

Related concepts:

- “Installation methods for DB2” in the *Quick Beginnings for DB2 Servers*

Related tasks:

- “Extending the directory schema (Windows 2000 and Windows .NET)” on page 20
- “Starting the DB2 Setup wizard (Windows)” on page 105
- “Applying the latest FixPak” on page 25
- “Verifying the installation using the command line processor (CLP)” on page 26
- “Installing DB2 online documentation (Windows)” on page 27

Related reference:

- “User accounts for installation and setup of DB2 Personal Edition” on page 104
- “Installation requirements for DB2 Personal Edition (Windows)” on page 100
- “Disk requirements for DB2 Personal Edition (Windows)” on page 102
- “Memory requirements for DB2 Personal Edition (Windows)” on page 102

DB2 Personal Edition installation overview (Windows)

This topic provides an installation overview for DB2[®] Personal Edition (Windows).

Installation overview:

Preparing your environment for installation

Before you install, you must prepare your computer for installation. To prepare your computer, you will:

1. Verify that your computer meets the necessary installation requirements.
2. Ensure that your system has enough memory to run DB2 Personal Edition.
3. Ensure that your system has enough space to install DB2 Personal Edition.
4. Ensure that you have the necessary user accounts for installation and setup. You require one user account for installation and two

user accounts for setup. The user accounts required for setup can be created before you install or you can have the DB2 Setup wizard create them for you.

5. If you are installing on Windows® 2000 or Windows .NET and are planning to use Light Weight Directory Access Protocol (LDAP), you will extend the Windows 2000 or Windows .NET directory schema so that it can contain DB2 object classes and attribute definitions.

Installing DB2

After preparing your environment, you will install DB2 Personal Edition using the DB2 Setup wizard. DB2 Setup wizard features include:

- A DB2 Setup Launchpad from which you can view installation notes, release notes, and learn about DB2 version 8 features
- Typical, Compact, and Custom installation types. Installation choices presented to you depend on the type of installation you choose
- The option to install support for multiple languages
- DB2 Administration Server setup (including DAS user setup)
- Administration contact and health monitor notification setup
- Instance setup and configuration (including instance user setup)
- DB2 tools catalog and warehouse control database setup
- Response file creation

Installing the latest FixPak

After you install DB2 using the DB2 Setup wizard, it is recommended that you install the latest DB2 version 8 FixPak. DB2 FixPaks are available on the IBM® support site.

Verifying the installation

After you install DB2 using the DB2 Setup wizard and have applied the latest DB2 FixPak, it is recommended that you verify the installation. To verify the installation, you will:

1. Create a sample database using the **db2sampl** command. You can also create a sample database using the First Steps utility, if you choose to install it.
2. Once the sample database has been created successfully, you will run SQL commands to retrieve sample data.

Related concepts:

- “Instance creation” in the *Administration Guide: Implementation*

Related tasks:

- “Installing DB2 Personal Edition (Windows)” on page 97
- “Initializing a warehouse control database during installation” in the *Data Warehouse Center Administration Guide*
- “Tools catalog database and DAS scheduler setup and configuration” in the *Administration Guide: Implementation*
- “Notification and contact list setup and configuration.” in the *Administration Guide: Implementation*

Related reference:

- “UPDATE HEALTH NOTIFICATION CONTACT LIST Command” in the *Command Reference*
- “db2setup - Install DB2 Command” in the *Command Reference*

Installation requirements for DB2 Personal Edition (Windows)

To install DB2 Personal Edition, the following operating system, software, and communications requirements must be met:

Operating system requirements

One of:

- Windows 98
- Windows ME
- Windows NT Version 4 with Service Pack 6a or higher
- Windows 2000 (32-bit)
- Windows XP (32-bit or 64-bit)
- Windows .NET (32-bit or 64-bit)

Windows XP (64-bit) and Windows .NET (64-bit) support:

- local 32-bit applications
- 32-bit UDFs and stored procedures

Hardware requirements

For DB2 products running on Intel and AMD systems, a Pentium or Athlon CPU is required.

Software requirements

- If you plan to use the Tivoli Storage Manager facilities for backup and restore of your databases, you require the Tivoli Storage Manager Client Version 3 or later.
- MDAC 2.7 is required. The DB2 Setup wizard will install MDAC 2.7 if it is not already installed.
- Java Runtime Environment (JRE) Version 1.3.1 is required to run DB2’s Java-based tools, such as the Control Center.

- A browser is required to view online help.

Communication requirements

- To connect to a remote database, you can use APPC, TCP/IP, and MPTN (APPC over TCP/IP). To remotely administer a version 8 DB2 database, you must connect using TCP/IP.
- To connect to a remote database using SNA (APPC), one of the following communication products is required:

Table 8. Windows SNA communications requirements

Operating system	SNA (APPC) communication product
Window ME	IBM Personal Communications Version 5.0 (CSD 3)
Windows NT	IBM Communications Server Version 6.1 or later, or IBM Personal Communications Version 5.0 (CSD 3)
Windows 2000	IBM Communications Server Version 6.1.1, or IBM Personal Communications Version 5.0 (CSD 3)
Windows XP	IBM Personal Communications Version 5.5
Windows .NET	SNA is not supported

- If you plan to use LDAP (Lightweight Directory Access Protocol), you require either a Microsoft LDAP client or an IBM SecureWay LDAP client V3.1.1.
- If you plan to use the Simple Network Management Protocol (SNMP) subagent, you require DPI 2.0 provided by IBM SystemView Agent. SNMP is not supported with DB2 offerings on Windows 64-bit platforms.
- Connections from 64-bit clients to downlevel 32-bit servers are not supported.
- Connections from downlevel 32-bit clients to 64-bit servers only support SQL requests.
- DB2 Version 8 Windows 64-bit servers support connections from DB2 Version 6 and Version 7 32-bit clients only for SQL requests. Connections from Version 7 64-bit clients are not supported.

Related tasks:

- “Installing DB2 Personal Edition (Windows)” on page 97

Memory requirements for DB2 Personal Edition (Windows)

The following table provides recommended memory requirements for DB2 Personal Edition installed with and without graphical tools. There are a number of graphical tools you can install including the Control Center, Configuration Assistant, and Data Warehouse Center.

Table 9. Memory requirements for DB2 Personal Edition

Type of installation	Recommended memory (RAM)
DB2 Personal Edition without graphical tools	64 MB
DB2 Personal Edition with graphical tools	128 MB

When determining memory requirements, be aware of the following:

- The memory requirements documented above do not account for non-DB2 software that may be running on your system.
- Memory requirements may be affected by the size and complexity of your database system.

Related tasks:

- “Installing DB2 Personal Edition (Windows)” on page 97

Disk requirements for DB2 Personal Edition (Windows)

The disk space required for DB2 Personal Edition depends on the type of installation you choose. The DB2 Setup wizard provides Typical, Compact, and Custom installation types. This table provides an approximate disk space requirements for each installation type.

Table 10. DB2 Personal Edition disk requirements

Installation type	required disk space
Typical	150 to 200 MB
Compact	100 to 150 MB
Custom	100 to 450 MB

Typical installation

DB2 Personal Edition is installed with most features and functionality, using a typical configuration. Typical installation includes graphical tools such as the Control Center and Configuration Assistant. You can also choose to install a typical set of data warehousing features.

Compact installation

Only the basic DB2 Personal Edition features and functions are installed. Compact installation does not include graphical tools.

Custom installation

A custom installation allows you to select the features you want to install.

The DB2 Setup wizard will provide a disk space estimate for the installation options you select.

Remember to include disk space allowance for required software, communication products, and documentation. In DB2 version 8, HTML and PDF documentation is provided on separate CD-ROMs.

Related tasks:

- “Installing DB2 Personal Edition (Windows)” on page 97

Extending the directory schema (Windows 2000 and Windows .NET)

If you plan to use LDAP with Windows 2000 or Windows .NET, you must extend the directory schema to contain DB2 object classes and attribute definitions. You must do this once before you install any DB2 products.

Prerequisites:

Your Windows user account must have Schema Administration authority.

Procedure:

To extend the directory schema:

1. Logon to a domain controller.
2. Run the **db2schex.exe** program from the installation CD with Schema Administration authority. You can run this program with Schema Administration authority, without logging off and logging on again, as follows:

```
runas /user:MyDomain\Administrator x:\db2\Windows\utilities\db2schex.exe
```

where x: represents the CD-ROM letter.

When **db2schex.exe** completes, you can continue with the installation.

Related reference:

- “Installation requirements for DB2 servers (Windows)” on page 17

User accounts for installation and setup of DB2 Personal Edition

If you are installing on Windows NT, Windows 2000, Windows XP, and Windows .NET, you require an installation user account and two user accounts for setup. The installation user account must be defined prior to running the DB2 Setup wizard. The setup user accounts (DB2 Administration server user, and DB2 instance user) can be defined prior to installation or you can have the DB2 Setup program create them for you.

All user account names must adhere to your system naming rules and to DB2 naming rules.

DB2 Personal Edition user accounts:

Installation user account

A local or domain user account is required to perform the installation. The user account must belong to the *Administrators* group on the machine where you will perform the installation and must have the following user rights:

- *Act as part of the operating system*

You can perform the installation without these user rights, but the installation program will be unable to validate accounts.

DB2 Administration Server user account

A local or domain user account is required for the DB2 Administration Server (DAS). You can create the DAS user account before installing DB2 or you can have the DB2 Setup wizard create it for you. The user account must belong to the *Administrators* group on the machine where you will perform the installation and must have the following user rights:

- *Act as part of the operating system*
- *Create token object*
- *Increase quotas*
- *Lock pages in memory*
- *Logon as service*
- *Replace a process level token*

The DB2 Administration Server (DAS) is a special DB2 administration service used to support the GUI tools and assist with administration tasks on local and remote DB2 servers. The DAS has an assigned user account that is used to log the DAS service on to the computer when the DAS service is started. It is recommended that the DAS user have SYSADM authority on each of the DB2 systems within your environment so that it can start or stop other instances if required.

DB2 instance user account

A local or domain user account is required for the DB2 instance. You can create the DB2 instance user account before installing DB2 or you can have the DB2 Setup wizard create it for you. The user account must belong to the *Administrators* group on the machine where you will perform the installation and must have the following user rights:

- *Act as part of the operating system*
- *Create token object*
- *Increase quotas*
- *Lock pages in memory*
- *Logon as service*
- *Replace a process level token*

On Windows, a DB2 instance exists as a service, except on Windows ME where a DB2 instance exists as a process. Every DB2 instance has one user that is assigned when the instance is created. DB2 logs on with this user name when the instance is started.

Related concepts:

- Appendix G, “User, userID and group naming rules” on page 379
- “DB2 system administrator group (Windows)” in the *Quick Beginnings for DB2 Personal Edition*

Related tasks:

- “Installing DB2 Personal Edition (Windows)” on page 97
- “Granting user rights (Windows)” in the *Quick Beginnings for DB2 Personal Edition*

Starting the DB2 Setup wizard (Windows)

Starting the DB2 Setup Wizard (Windows) is part of the larger task of *Installing DB2 (Windows)*.

This task describes how to start the DB2 Setup wizard on Windows. You will use the DB2 Setup wizard to define your installation and install DB2 to your system.

Prerequisites:

Before you start the DB2 Setup wizard:

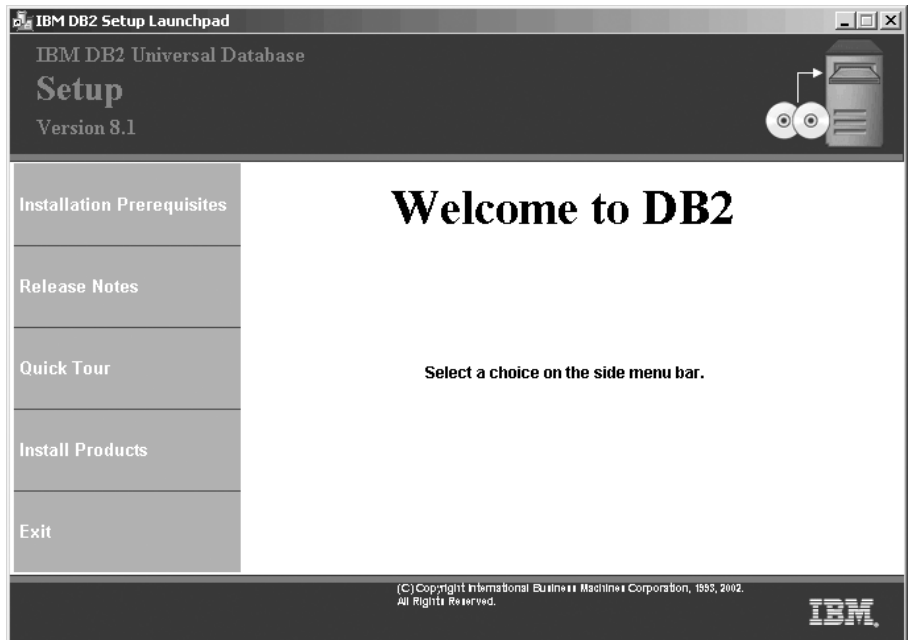
- Ensure that your system meets installation, memory, and disk requirements.

- If you are planning to use LDAP on Windows 2000 or Windows .NET, you must extend the directory schema before you install.
- You must have an account with local administrative privileges and the recommended user rights to perform the installation.

Procedure:

To start the DB2 Setup wizard:

1. Log on to the system with the Administrator account that you have defined for DB2 installation.
2. Close all programs so the installation program can update files as required.
3. Insert the CD-ROM into the drive. If enabled, the auto-run feature automatically starts the DB2 Setup launchpad:



From this window, you can view installation prerequisites and the release notes, you can take the DB2 Quick Tour to explore the features of DB2 Universal Database Version 8, or you can proceed directly to the installation. You may want to review the installation prerequisites and release notes for late-breaking information. Select **Install Products** and select the DB2 product to install.

4. The DB2 Setup wizard will determine the system language, and launch the setup program for that language. If you want to run the setup program in a different language, or the setup program failed to auto-start, you can

start the DB2 Setup wizard manually. The syntax for starting the DB2 Setup wizard is described at the end of this procedure.

5. Once you have initiated the installation, proceed by following the setup program's prompts. Online help is available to guide you through the remaining steps. To invoke the online help, click Help or press F1. You can click **Cancel** at any time to end the installation. DB2 files will only be copied to your system once you have clicked **Finish** on the last DB2 Setup wizard installation panel.

For information on errors encountered during installation, see the `db2.log` file. The `db2.log` file stores general information and error messages resulting from the install and uninstall activities. By default, the `db2.log` file is located in the `x:\My Documents\DB2LOG\` directory, where `x`: represents the drive on which your operating system is installed.

To start the DB2 Setup wizard manually:

1. Click **Start** and select the **Run** option.
2. In the **Open** field, enter the following command:

```
x:\setup /i language
```

where:

- `x`: represents your CD-ROM drive
- `language` is the territory identifier for your language (for example, EN for English).

The `/i language` parameter is optional. If it is not specified, the DB2 Setup wizard will run in the same language as your operating system.

3. Click **OK**.

Your next step is *Applying the latest FixPak*.

Related tasks:

- “Extending the directory schema (Windows 2000 and Windows .NET)” on page 20

Related reference:

- “User accounts for installation and setup of DB2 Personal Edition” on page 104
- “Installation requirements for DB2 Personal Edition (Windows)” on page 100
- “Disk requirements for DB2 Personal Edition (Windows)” on page 102
- “Memory requirements for DB2 Personal Edition (Windows)” on page 102

Applying the latest FixPak

Applying the latest FixPak is optionally part of the larger task of installing DB2 products.

A DB2 FixPak contains updates and fixes for bugs (Authorized Program Analysis Reports, or "APARs") found during testing at IBM, as well as fixes for bugs reported by customers. Every FixPak is accompanied by a document, called APARLIST.TXT, that describes the bug fixes it contains.

FixPaks are cumulative. This means that the latest FixPak for any given version of DB2 contains all of the updates from previous FixPaks for the same version of DB2. We recommend that you keep your DB2 environment running at the latest FixPak level to ensure problem-free operation.

When installing a FixPak on a partitioned ESE system, all participating computers must have the same FixPak installed while the system is offline.

Prerequisites:

Each FixPak may have specific prerequisites. See the FixPak README that accompanies the FixPak for more information.

Procedure:

1. Download the latest DB2 FixPak from the IBM DB2 UDB and DB2 Connect Online Support Web site at <http://www.ibm.com/software/data/db2/udb/winos2unix/support>.
2. Each FixPak contains a set of Release Notes and a README. The README provides instructions for installing the FixPak.

Verifying the installation using the command line processor (CLP)

Verifying the installation using the command line processor (CLP) is optionally part of the larger task of *Installing DB2*.

Once you have completed installing DB2, you can verify the installation by creating a sample database and running SQL commands to retrieve sample data.

Prerequisites:

- The Sample Database component must be installed on your system. The Sample Database component is included in a typical installation.
- You require a user with SYSADM authority.

Procedure:

To verify the installation:

1. Log on to the system as a user with SYSADM authority.
2. Enter the **db2sampl** command to create the SAMPLE database.
This command may take a few minutes to process. There is no completion message; when the command prompt returns, the process is complete.
The SAMPLE database is automatically cataloged with the database alias SAMPLE when it is created.
3. Start the database manager by entering the **db2start** command.
4. Enter the following DB2 commands from a DB2 command window to connect to the SAMPLE database, retrieve a list of all the employees that work in department 20, and reset the database connection:

```
db2 connect to sample
db2 "select * from staff where dept = 20"
db2 connect reset
```

After you have verified the installation, you can remove the SAMPLE database to free up disk space. Enter the **db2 drop database sample** command to drop the SAMPLE database.

Related tasks:

- “Verifying the installation of DB2 servers using First Steps” in the *Quick Beginnings for DB2 Servers*

Installing DB2 online documentation (Windows)

This task describes how to install the DB2 online documentation using the DB2 Setup wizard on Windows. The DB2 online documentation is installed separately from other DB2 products from its own CD-ROM.

Prerequisites:

Before you start the DB2 Setup wizard:

- Ensure that your system meets installation, memory, and disk requirements.
- You must have a local *Administrator* user account with the recommended user rights to perform the installation.

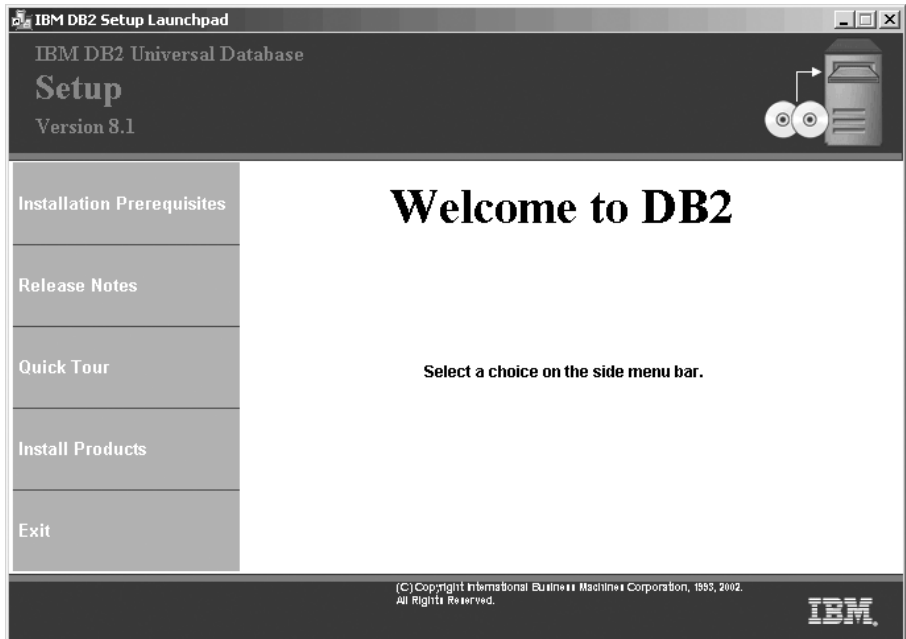
Procedure:

To start the DB2 Setup wizard:

1. Insert the CD-ROM into the drive. The auto-run feature automatically starts the DB2 Setup wizard. The DB2 Setup wizard will determine the

system language, and launch the setup program for that language. If you want to run the setup program in a different language, or the setup program failed to auto-start, you can start the DB2 Setup wizard manually.

2. The DB2 Setup Launchpad opens.



From this window, you can view installation prerequisites and the release notes, you can take a Quick Tour to explore the features of DB2 Universal Database Version 8, or you can proceed directly to the installation. You may want to review the installation prerequisites and release notes for late-breaking information.

3. Once you have initiated the installation, proceed by following the setup program's prompts. Online help is available to guide you through the remaining steps. To invoke the online help, click Help or press F1. You can click **Cancel** at any time to end the installation. DB2 files will only be copied to your system once you have clicked **Finish** on the last DB2 Setup wizard installation panel.

For information on errors encountered during installation, see the db2.log file. The db2.log file stores general information and error messages resulting from the install and uninstall activities. By default, the db2.log file is located in the 'My Documents'\DB2LOG\ directory. The location of the 'My Documents' directory will depend on the settings on your computer.

To start the DB2 Setup wizard manually:

1. Click **Start** and select the **Run** option.

2. In the **Open** field, enter the following command:

```
x:\setup /i language
```

where:

- *x*: represents your CD-ROM drive
- *language* is the territory identifier for your language (for example, EN for English).

The /i language parameter is optional. If it is not specified, the DB2 Setup wizard will run in the same language as your operating system.

3. Click **OK**.

Chapter 4. Performing a Response File Installation

Response file installation types	113	Making DB2 files available for a response file installation	130
Response files	114	Setting up shared access to a directory on Windows	131
Available sample response files	115	Creating a response file on Windows	132
Response file keywords	116	Running setup with the response file from the client workstation on Windows	133
Response file keywords	121	Installing DB2 products using Microsoft Systems Management Server (SMS)	135
DB2 Control Server response file keywords for Windows operating systems	122	Importing the DB2 install file into SMS	136
Response file generator	123	Creating the SMS package on the SMS server	137
db2rspgn - Response file generator	124	Distributing the DB2 installation package across your network	138
Killing DB2 processes during an interactive installation	125	Configuring remote access to a server database	140
Killing DB2 processes during a response file installation	126	Configuring db2cli.ini for a response file installation	142
Response file installation of DB2 on UNIX	126	Exporting and importing a profile	142
Creating a response file on UNIX	128		
Performing a response file installation on UNIX	129		
Response file installation of DB2 on Windows	129		

Response file installation types

DB2[®] products can be installed non-interactively using response files. This installation type can be used with systems management software such as Microsoft[®] Systems Management Server (SMS) or simply with a shared CD-ROM or network drive.

To decrease the amount of time it will take to perform the installation, you should copy the contents of the CD to a directory on your machine, and perform the install from there.

Related concepts:

- “Response files” on page 114

Related tasks:

- “Installing DB2 products using Microsoft Systems Management Server (SMS)” on page 135
- “Response file installation of DB2 on UNIX” on page 126
- “Response file installation of DB2 on Windows” on page 129
- “Distributing the DB2 installation package across your network” on page 138

Response files

The first step in any type of distributed installation is the creation of a response file. A response file is an ASCII file that can be customized with the setup and configuration data that will automate an installation. The setup and configuration data would have to be entered during a interactive install, but with a response file, the installation can proceed without any intervention.

A response file specifies configuration and setup parameters such as the destination directory and the products and components to install. It can also be used to set up the following settings:

- Global DB2[®] registry variables
- Instance variables
- Instance database manager configuration settings

You can create a response file by:

- Modifying the sample response files that are provided.
- Using the response file generator for Windows[®] systems.
- Using the DB2 Setup Wizard.

The following is a list of response files considerations:

- The created response file can be accessed by running through the GUI portion of the install.
- To use the response file generator, the install process must be completed.
- The response file created using the DB2 Setup wizard can be used to install other nodes in an ESE partitioned configuration. For example, you can customize a sample response file which will install a DB2 Administration Client by using the DB2 Setup Wizard which is a new feature of V8.
- You can install a product and create a response file for installing the same product on another system or you can run the DB2 Setup wizard and create the response file only.

You can use a response file to install an identical configuration across every workstation on your network or to install multiple configurations of a DB2 product. You can then distribute this file to every workstation where you want this product to be installed.

Related concepts:

- “Response file generator” on page 123

Related reference:

- “Response file keywords” on page 116

- “DB2 Control Server response file keywords for Windows operating systems” on page 122
- “db2rspgn - Response file generator” on page 124

Available sample response files

The DB2 CD-ROM includes a ready-to-use sample response files with default entries. The sample response files are located in:

`db2/platform/samples`

where *platform* is one of the following:

- `hpux`
- `aix`
- `solaris`
- `linux`
- `linux64`
- `linux390`
- `windows`

You can use the following sample response files to install DB2 products on supported workstations:

db2adcl.rsp	DB2 Application Development Client
db2admcl.rsp	DB2 Administration Client
db2conee.rsp	DB2 Connect Enterprise Edition
db2conpe.rsp	DB2 Connect Personal Edition
db2dlm.rsp	DB2 Data Links Manager
db2ese.rsp	DB2 Enterprise Server Edition
db2gse.rsp	DB2 Spatial Extender Server
db2lsdc.rsp	DB2 Life Sciences Data Connect
db2pe.rsp	DB2 Personal Edition
db2rcon.rsp	DB2 Relational Connect
db2rtcl.rsp	DB2 Run-Time Client
db2wm.rsp	DB2 Warehouse Manager
db2wmc.rsp	DB2 Warehouse Manager Connectors
db2wse.rsp	DB2 Workgroup Server Edition

Related concepts:

- “Response files” on page 114

Related reference:

- “Response file keywords” on page 116
- “DB2 Control Server response file keywords for Windows operating systems” on page 122

Response file keywords

This section describes some of the keywords that you will specify when performing a distributed installation. You can use the response file to install additional components/products after an initial install.

PROD Specifies the product that you want to install. The options are:

- ADMINISTRATION_CLIENT for the DB2 Administration Client
- APPLICATION_DEVELOPMENT_CLIENT for the DB2 Application Development Client
- CONNECT_PERSONAL_EDITION for DB2 Connect Personal Edition
- CONNECT_ENTERPRISE_EDITION for DB2 Connect Enterprise Edition
- DATA_LINKS_MANAGER for DB2 Data Links Manager
- DB2_HTML_DOCUMENTATION for the DB2 HTML Documentation CD
- ENTERPRISE_SERVER_EDITION for DB2 Enterprise Server Edition
- LIFE_SCIENCES_DATA_CONNECT for DB2 Life Sciences Data Connect
- PERSONAL_EDITION for DB2 Personal Edition
- RELATIONAL_CONNECT for DB2 Relational Connect
- RUNTIME_CLIENT for the DB2 Run-Time Client
- SPATIAL_EXTENDER for DB2 Spatial Extender Server
- WAREHOUSE_MANAGER for DB2 Data Warehouse Manager
- WAREHOUSE_MANAGER_CONNECTORS for the DB2 Data Warehouse Manager Connectors
- WORKGROUP_SERVER_EDITION for DB2 Workgroup Server Edition

Note: You should not comment out the **PROD** keyword as you may have some missing components even with a successful response file installation.

FILE Specifies the destination directory for a DB2 product.

Note: **FILE** is for Windows only.

INSTALL_TYPE

Specifies the type of install.

The options are:

- COMPACT
- TYPICAL
- CUSTOM

Important: A compact or typical install type will ignore any custom keywords (COMP).

TYPICAL_OPTION

A typical install contains function applicable for most users of the product. The TYPICAL options adds to this functionality by installing additional functionality that is typical for users installing either a data warehousing environment or a satellite environment. These options are only valid if the INSTALL_TYPE keyword is equal to TYPICAL. For example, remove the * (uncomment) from the following:

```
*TYPICAL_OPTION = DATA_WAREHOUSE
*TYPICAL_OPTION = SATALLITE_ADMIN
```

COMP

Specifies the components that you want to install. The setup program automatically installs components that are required for a product, and ignores requested components that are not available.

In a custom install, you must select components individually. This can be done by uncommenting the COMP keywords for component that you want installed (this differs depending on the product). For example, to install the CA, remove the * (uncomment) from the following:

```
*COMP = CONFIGURATION_ASSISTANT
```

Note: This keyword is ignored unless your **INSTALL_TYPE** is *CUSTOM*.

LANG

This refers to language selection keywords. You must uncomment any additional languages that you would like to install. The English language is mandatory and is always selected. For example, to install French, remove the * (uncomment) from the following:

```
*LANG=FR
```

REBOOT

Specifies whether to restart the system when the installation has completed.

Note: REBOOT is for Windows only.

KILL_PROCESSES

If you have an existing version of DB2 and it is running and this keyword is set to YES, it will terminate your running DB2 processes without prompt.

Note: KILL_PROCESSES is for Windows only.

DB2 Administration Server settings

To enable any of the following DAS settings, remove the * (uncomment). This setting is applicable for both Windows and UNIX environments:

- On UNIX:
 - *DAS_USERNAME = dasuser
 - *DAS_PASSWORD = dasp
 - *DAS_GID = 100
 - *DAS_UID = 100
 - *DAS_GROUP_NAME = dasgroup
 - *DAS_SMTP_SERVER = jsmith.torolab.ibm.com
- On Windows:
 - *DAS_USERNAME = dasuser
 - *DAS_DOMAIN = domain
 - *DAS_PASSWORD = dasp
 - *DAS_SMTP_SERVER = jsmith.torolab.ibm.com

The options below specify where the DAS contact list will be kept. If the contact list is remote, then you must specify a username and password that has authority to add a contact to the system.

```
*DAS_CONTACT_LIST = LOCAL or REMOTE (DEFAULT = LOCAL)
*DAS_CONTACT_LIST_HOSTNAME = hostname
*DAS_CONTACT_LIST_USERNAME = username
*DAS_CONTACT_LIST_PASSWORD = password
```

Special instance specifications

All of these take instance sections not instance names. The instance section must exist in the response file.

- Windows:
 - DEFAULT_INSTANCE - This is the default instance.
 - CTLSRV_INSTANCE - This is the instance that is configured to act as the satellite control server.
- UNIX:
 - UNIX_WAREHOUSE_INSTANCE - This tells the install which instance will be set up to use Data Warehouse. The IWH.environment file will be updated with the name of the instance whose section appears here.

Instance specifications

You can use the response file to create as many instances as you want.

To create a new instance you must specify an instance section using the `INSTANCE` keyword. Once this has been done, any keywords that contain the value specified in `INSTANCE` as a prefix belong to that instance. The following are examples of instance specifications for both Windows and UNIX environments:

- On UNIX:

```
*INSTANCE=DB2_INSTANCE
*DB2_INSTANCE.NAME = db2inst1
*DB2_INSTANCE.TYPE = ESE
*DB2_INSTANCE.PASSWORD = PASSWORD
*DB2_INSTANCE.UID = 100
*DB2_INSTANCE.GID = 100
*DB2_INSTANCE.GROUP_NAME = db2grp1
*DB2_INSTANCE.HOME_DIRECTORY = /home/db2inst1
*DB2_INSTANCE.SVCENAME = db2cdb2inst1
*DB2_INSTANCE.PORT_NUMBER = 50000
*DB2_INSTANCE.FCM_PORT_NUMBER = 60000
*DB2_INSTANCE.MAX_LOGICAL_NODES = 4
*DB2_INSTANCE.AUTOSTART = YES
*DB2_INSTANCE.DB2COMM = TCPIP
*DB2_INSTANCE.WORDWIDTH = 32
*DB2_INSTANCE.FENCED_USERNAME = USERNAME
*DB2_INSTANCE.FENCED_PASSWORD = PASSWORD
*DB2_INSTANCE.FENCED_UID = 100
*DB2_INSTANCE.FENCED_GID = 100
*DB2_INSTANCE.FENCED_GROUP_NAME = db2grp1
*DB2_INSTANCE.FENCED_HOME_DIRECTORY = /home/db2inst1
```

- On Windows:

```
*INSTANCE = DB2_INSTANCE
*DB2_INSTANCE.NAME = db2inst1
*DB2_INSTANCE.TYPE = ESE
*DB2_INSTANCE.PASSWORD = PASSWORD
*DB2_INSTANCE.USERNAME = db2admin
*DB2_INSTANCE.SVCENAME = db2cdb2inst1
*DB2_INSTANCE.PORT_NUMBER = 50000
*DB2_INSTANCE.FCM_PORT_NUMBER = 60000
*DB2_INSTANCE.MAX_LOGICAL_NODES = 4
*DB2_INSTANCE.AUTOSTART = YES
*DB2_INSTANCE.DB2COMM = TCPIP, NETBIOS, NPIPE
```

Database Section

These keywords can be used to have the install create or catalog a database on the machine that is being installed.

```
DATABASE = DATABASE_SECTION
DATABASE_SECTION.INSTANCE = DB2_INSTANCE
DATABASE_SECTION.DATABASE_NAME = TOOLSDB
DATABASE_SECTION.LOCATION = LOCAL
DATABASE_SECTION.ALIAS = TOOLSDB
DATABASE_SECTION.USERNAME = username
DATABASE_SECTION.PASSWORD = password
```

* these keywords are only used for REMOTE databases

that are being cataloged
DATABASE_SECTION.SYSTEM_NAME = *hostname*
DATABASE_SECTION.SVCENAME = db2cdb2inst1

WAREHOUSE_CONTROL_DATABASE

The value for this keyword should be one of the Database Section keywords that were specified in the response file. For example:

```
*WAREHOUSE_CONTROL_DATABASE = DATABASE_SECTION
```

. The database section that is specified with this keyword must specify the USERNAME and PASSWORD keywords.

WAREHOUSE_SCHEMA

For example, to set the warehouse schema, remove the * (uncomment) from the following:

```
*WAREHOUSE_SCHEMA = wm_schema
```

ICM_DATABASE

This specifies the database to use to store the information catalog. The value for this keyword should be one of the Database Section keywords that were specified in the response file.

```
*ICM_DATABASE = DATABASE_SECTION
```

ICM_SCHEMA

For example, to set the information catalog schema, remove the * (uncomment) from the following:

```
*ICM_SCHEMA = icm_schema
```

TOOLS_CATALOG_DATABASE

This specifies the database to use to store the tools catalog. The value for this keyword should be one of the Database Section keywords that were specified in the response file.

```
*TOOLS_CATALOG_DATABASE = DATABASE_SECTION
```

TOOLS_CATALOG_SCHEMA

For example, to set the tools catalog schema, remove the * (uncomment) from the following:

```
*TOOLS_CATALOG_SCHEMA = toolscat_schema
```

SATELLITE_CONTROL_DATABASE

This section specifies the database you would like to use as the satellite Control Server. The value for this keyword should be one of the Database Section keywords that were specified in the response file:

```
*SATELLITE_CONTROL_DATABASE = DATABASE_SECTION
```

Contact Section

These keywords define a contact section that will be created by the

installation process if it does not already exist and the Health notifications for the instance that is specified will be sent to this contact.

```
CONTACT = contact_section
contact_section.CONTACT_NAME = contact name
contact_section.INSTANCE = DB2_INSTANCE
contact_section.EMAIL = Email address
contact_section.PAGER = NO
```

Related concepts:

- “Response files” on page 114

Related reference:

- “Available sample response files” on page 115

Response file keywords

This section describes the most important keywords that you will specify when performing a distributed installation of DB2 Universal Database on Windows 32-bit operating systems.

COMP

Specifies the components you want to install. The setup program automatically installs components that are required for a product, and ignores requested components that are not available.

Note: Component selections have no effect unless you specify a custom installation (TYPE = 2).

DB2.AUTOSTART

Specifies whether or not to automatically start the DB2 instance each time the system is rebooted.

By default, the DB2 instance starts automatically unless this parameter is set to NO.

DB2.DB2SATELLITEID

Specifies the unique ID for the satellite and sets the DB2SATELLITEID registry variable on the satellite. The ID must be unique across all groups that are recorded at the satellite control server. It must match an ID defined for a satellite at the control server. The satellite ID is used during the synchronization process to identify the satellite. The ID can be a maximum of 20 characters.

It is not recommended that you provide the DB2SATELLITEID in the response file as it must be unique, unless you are customizing the

value of DB2SATELLITEID for each system the response file will be used on. The DB2SATELLITEID can be set after the installation using the **db2set** command.

If not specified, the Windows login ID is used in its place during the synchronization process.

DB2.DB2SATELLITEAPPVER

Specifies the satellite's application software version. The version can be a maximum of 18 characters and numbers. The value specified must match an application version defined for the group to which the satellite belongs as defined at the satellite control server. If it does, then scripts associated with this application version will be used to maintain the satellite during the synchronization process. A default version V1ROM00 has been supplied, but this value can be changed. These values can be set or changed after installation.

DB2SYSTEM

Specifies a name for the system which is unique within a network. This parameter must be specified.

FILE Specifies the destination directory for a DB2 product.

PROD Specifies the product you want installed. You want to install UDB_SATELLITE for DB2 Universal Database.

REBOOT

Specifies whether to restart the system when the installation is complete.

TYPE Specifies the type of install.

The options are:

- Typical
- Compact
- Custom

Note: A compact or typical install type will ignore any custom keywords (COMP).

DB2 Control Server response file keywords for Windows operating systems

This section describes some of the keywords that you will specify when performing a distributed installation of the DB2 Control Server on Windows operating systems (Windows NT, Windows 2000, Windows XP, and Windows .NET). The DB2 Control Server provides administrative and status reporting support for satellites by using the satellite control database SATCTLDB. This database is automatically created when the Control Server component is

installed. These keywords can be used to specify the values of database manager configuration parameters and the values of the DB2 registry variables.

To install the Control Server, select the CONTROL_SERVER component (COMP=CONTROL_SERVER), which is only available on ESE.

CTLSRV.AUTOSTART

Specifies whether or not to automatically start the DB2 Control Server instance (DB2CTLSV) each time the system is rebooted.

The default is YES, the DB2CTLSV instance starts automatically.

CTLSRV.SVCENAME

Specifies the DB2 Control Server instance, TCP/IP service name and can be used to override the default service name generated by the installation program. When used in conjunction with the CTLSRV.PORT_NUMBER keyword to override the default port number, you have complete control over the TCP/IP configuration for the DB2 Control Server instance.

CTLSRV.PORT_NUMBER

Specifies the DB2 Control Server instance, TCP/IP service name and can be used to override the default service name generated by the installation program. When used in conjunction with the CTLSRV.SVCENAME keyword to override the default port number, you have complete control over the TCP/IP configuration for the DB2 Control Server instance.

Related concepts:

- “Response files” on page 114

Related reference:

- “Available sample response files” on page 115

Response file generator

The response file generator utility, which is available on Windows® 32-bit and 64-bit operating systems, creates a response file from an existing installed and configured DB2® product. You can use the generated response file to recreate the exact setup on other machines.

For example, you could install and configure a DB2 Run-Time client to connect to various databases across your network. Once this DB2 client is installed and configured to access all the databases that your users to have access to, you can run the response file generator to create a response file and a profile for each instance.

The response file generator creates a response file for the installation and instance profiles for each instance that you specify. You can then use the response file to create identical clients across your network.

The response file generator also gives you the option to create the installation response file without an instance profile. This would allow you to create identical copies of your installed client without the configuration information.

Related tasks:

- “Response file installation of DB2 on UNIX” on page 126
- “Response file installation of DB2 on Windows” on page 129
- “Distributing the DB2 installation package across your network” on page 138

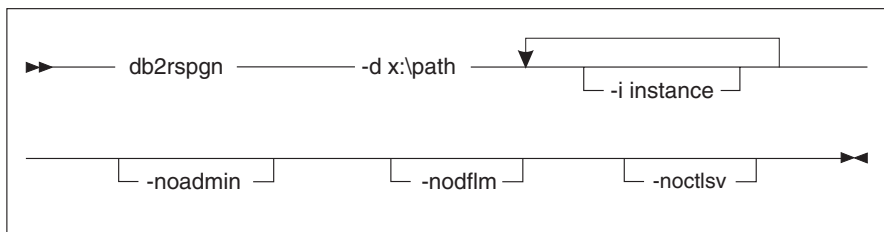
Related reference:

- “db2rspgn - Response file generator” on page 124

db2rspgn - Response file generator

The response file generator is available on Windows only.

The syntax for the **db2rspgn** command is as follows:



- d** Destination directory for a response file and any instance files. This parameter is required.
- i** A list of instances for which you want to create a profile. The administration instance (DB2DAS00) does not need to be specified. The default is to generate an instance profile file for all instances. This parameter is optional.
- noadmin** Disables the saving of the administration instance (DB2DAS00). The administration instance will then be created with the standard defaults. The default is to save the administration instance. This parameter is optional.

-noctlsrv

Indicates that an instance profile file will not be generated for the Control Server instance. This parameter is optional.

Related concepts:

- “Response file generator” on page 123

Killing DB2 processes during an interactive installation

If any DB2 processes are running when the DB2 setup command is issued, the installation of DB2 cannot occur. For example, during an interactive installation, the following message is issued: DB2 is currently running and locked by the following process(es). The user is then prompted to kill the DB2 processes so that the installation can proceed. You should exercise extreme caution when you kill active DB2 processes so that an installation can occur. The termination of a DB2 process can cause the loss of data. The following describes how to kill these processes.

Restrictions:

The ability to specify that any running DB2 processes are killed when the DB2 setup command is issued is available on Windows 32-bit operating systems only. This process is not a necessary step on UNIX to perform an installation.

Procedure:

To kill any running DB2 processes for an interactive installation, specify the **/F** option for the setup command. The **/F** option kills the running processes, and the message and prompt are not displayed.

In addition, DB2 services can be viewed in the Services Window to ensure that they have been stopped.

Note: We recommend issuing the **db2stop** command for each instance before installing to lessen the risk of data loss.

Related tasks:

- “Killing DB2 processes during a response file installation” on page 126

Related reference:

- “db2stop - Stop DB2 Command” in the *Command Reference*

Killing DB2 processes during a response file installation

If any DB2 processes are running when the DB2 setup command is issued, the installation of DB2 cannot occur. The user must kill the DB2 processes so that the installation can proceed. You should exercise extreme caution when you kill active DB2 processes so that an installation can occur. The termination of a DB2 process can cause the loss of data. The following describes how to kill these processes.

Restrictions:

The ability to specify that any running DB2 processes are killed when the DB2 setup command is issued is available on Windows 32-bit and 64-bit operating systems only. This process is not a necessary step on UNIX to perform an installation.

Procedure:

For a response file installation, you can use either of the following methods to kill any active DB2 processes. If you specify either of these options, the active DB2 processes are killed before the installation proceeds.

- Specify the `/F` option for the setup command. You can use this option along with the `/U`, `/L` and `/I` options that are already available.
- Set the `KILL_PROCESSES` keyword to `YES` (the default is `N0`).

Note: We recommend issuing the `db2stop` command for each instance before installing to lessen the risk of data loss.

Related tasks:

- “Killing DB2 processes during an interactive installation” on page 125

Related reference:

- “`db2stop` - Stop DB2 Command” in the *Command Reference*

Response file installation of DB2 on UNIX

This task describes how to perform response file installations on UNIX. You can use the response file to install additional components/products after an initial install.

Restrictions:

You should be aware of the following limitations when using the response files method to install DB2 on UNIX platforms:

- If you set any instance or global profile registry keywords to BLANK (the word "BLANK"), the effect is to delete that keyword from the list of currently set keywords. If the registry variable corresponding to a keyword is not already set, and you run a response file install with this keyword set to BLANK, the installation will fail.
- If you are using the response file to install on Linux, please ensure that you have sufficient space prior to installing. Otherwise you may need to do some manual cleanup (such as removing RPMs that may be partially installed) if the installation fails.
- We recommend installing from a file system network drive rather than a CD-ROM drive. Installing from a network drive will significantly decrease the amount of time it will take to perform the install. If you are planning on installing multiple clients, you should set up a mounted file system on a code server to improve performance.

Prerequisites:

Before you begin the installation, ensure that:

- Your system meets all of the memory, hardware, and software requirements to install your DB2 product.
- For systems using NIS, you must set up all of the userids/groups before running the response file installation.

Procedure:

- Mount the CD-ROM for one of the following platforms:
 - Mounting the CD-ROM on AIX
 - Mounting the CD-ROM on HP-UX
 - Mounting the CD-ROM on Linux
 - Mounting the CD-ROM on Solaris
- Create a response file on UNIX
- Perform an unattended installation

Related tasks:

- "Mounting the CD-ROM on AIX" in the *Installation and Configuration Supplement*
- "Mounting the CD-ROM on HP-UX" in the *Installation and Configuration Supplement*
- "Mounting the CD-ROM on Linux" in the *Installation and Configuration Supplement*
- "Mounting the CD-ROM on Solaris" in the *Installation and Configuration Supplement*
- "Creating a response file on UNIX" on page 128

- “Performing a response file installation on UNIX” on page 129
- “Response file installation of DB2 on Windows” on page 129

Creating a response file on UNIX

Creating a response file is part of the larger task of *Response file installation of DB2 on UNIX*. The DB2 CD-ROM includes a ready-to-use sample response file with default entries. The sample response files are located in

```
<cd-rom>/db2/platform/samples
```

where <cd-rom> represents the location of the installable version of DB2.

Sample response files are available for each DB2 product.

Procedure:

To create a customized response file from the sample, perform the following steps:

1. Copy the sample response file to a local file system and edit it.
2. To activate an item in the response file, remove the asterisk (*) to the left of the keyword. Then, replace the current setting to the right of the value with the new setting. The possible settings are listed to the right of the equal sign.

Keywords that are unique to installation are only specified in a response file during a response file installation.

3. Save the file on an exported file system available to everyone on the network.

If you are installing directly from the CD-ROM, you must store the renamed response file on another drive.

Important: You can specify the name of the instance owner in the response file. If this user does not already exist, DB2 will create this user on your system.

Your next step is to perform a response file installation on UNIX.

Performing a response file installation on UNIX

Performing a response file installation is part of the larger task of *Response file installation of DB2 on UNIX*.

Prerequisites:

Before you begin the installation, ensure that:

- Installation of DB2 products using a response file requires that you have root authority.
- Refer to the installation documentation for the particular DB2 product that you want to install. For example, if you want to install DB2 Enterprise Server Edition, you must refer to the installation documentation for DB2 Enterprise Server Edition to review installation prerequisites, information about required users, and other important setup information.

Procedure:

To perform a response file installation, perform the following steps:

1. Enter the **db2setup** command as follows:

```
<cd-rom>/db2setup -r <responsefile_directory>/<response_file>
```

where:

- `<cd-rom>` represents the location of the DB2 installable image;
 - `<responsefile_directory>` represents the directory where the customized response file is located; and
 - `<response_file>` represents the name of the response file.
2. Check the messages in the log file when the installation finishes. The location of the log file is: `/tmp/db2setup.log`

Response file installation of DB2 on Windows

This section describes how to perform a response file installation on Windows.

Prerequisites:

Before you begin the installation, ensure that:

1. Your system meets all of the memory, hardware, and software requirements to install your DB2 product.
2. You have all of the required user accounts to perform the installation.

Procedure:

To distribute a response file installation of DB2:

- Make DB2 Files available for installation
- Set up shared access to a directory
- Create response files
- Run setup with the response file from the client workstation

Related tasks:

- “Making DB2 files available for a response file installation” on page 130
- “Setting up shared access to a directory on Windows” on page 131
- “Creating a response file on Windows” on page 132
- “Running setup with the response file from the client workstation on Windows” on page 133
- “Configuring db2cli.ini for a response file installation” on page 142
- “Installing DB2 products using Microsoft Systems Management Server (SMS)” on page 135
- “Response file installation of DB2 on UNIX” on page 126

Related reference:

- “Memory requirements for partitioned DB2 servers (UNIX)” on page 71
- “Installation requirements for DB2 servers (Windows)” on page 17

Making DB2 files available for a response file installation

Making DB2 files available for a response file installation is part of the larger task of a *Response file installation of DB2 on Windows*.

Prerequisites:

To make DB2 files available for a response file installation, you must copy the required files from the CD-ROM to the shared network drive that will act as the install server.

Procedure:

To copy the required files from the CD-ROM to the shared network drive that will act as the install server, perform the following steps:

1. Insert the appropriate CD-ROM into the drive.
2. Create a directory by entering the following command:

```
md c:\db2prods
```


3. Enter the **cpyssetup.bat** command to copy the DB2 installation files to your install server. This command is located in the `x:\db2\windows` directory, where *x*: represents your CD-ROM drive.

The command syntax is as follows:

```
cpyssetup.bat directory language
```

where:

- *directory* represents the directory that you created in the previous step (for example, `c:\db2prods`).
- *language* represents the two-character country/region code for your language (for example, `en` for English).

For example, to copy all of the English DB2 install files to the `c:\db2prods` directory, enter the following command:

```
cpyssetup.bat c:\db2prods en
```

Related tasks:

- “Setting up shared access to a directory on Windows” on page 131

Setting up shared access to a directory on Windows

Setting up shared access to a directory is part of the larger task of a *Response file installation of DB2 on Windows*. This task will allow you to grant your network workstations access to a directory on the code server.

Procedure:

To set up shared access to a directory on the code server:

1. Open Windows Explorer.
2. Select the directory that you want to share. For example, `c:\db2prods`.
3. Select **File**—>**Properties** from the menu bar. The properties window for the directory will open.
4. Select the **Sharing** tab.
5. Select the **Shared As** radio button.
6. In the **Share Name** field, enter a share name. For example, `db2nt`.
7. To specify *Read access* for everyone:
 - a. Click the **Permissions** push button. The Access Through Share Permissions window opens.
 - b. Ensure that the **Everyone** option is selected in the **Name** box.
 - c. Click the **Type of Access** drop down box and select the **Read** option.

- d. Click **OK**. You are returned to the properties window of the directory for which you want to set up shared access.
- e. Click **OK**.

Your next step is to create a response file on Windows.

Related tasks:

- “Creating a response file on Windows” on page 132
- “Making DB2 files available for a response file installation” on page 130

Creating a response file on Windows

Creating a response file is part of the larger task of a *Response file installation of DB2 on Windows*.

If you have already set up and configured a DB2 product and you want to distribute this exact configuration across your network, we recommend that you use the response file generator to create the response file for your installation.

The DB2 CD-ROM includes a ready-to-use sample response file with default entries. The sample response files are located in `x:\db2\windows\samples` directory, where `x`: represents the CD-ROM drive.

Response files are available for each DB2 product.

Procedure:

To edit the appropriate sample response file, perform the following steps:

1. Customize the response file.

To activate an item in the response file, remove the asterisk (*) to the left of the keyword. Then, replace the current setting to the right of the value with the new setting. The possible settings are listed to the right of the equal sign.

Some product response files have mandatory keywords that you must provide values for. The mandatory keywords are documented in the comments of each response file.

Keywords that are unique to installation are only specified in a response file during a response file installation.

2. Save the file. If you have made any changes, save the file under a new file name to preserve the original sample response file. If you are installing directly from the CD-ROM, you should store the renamed response file on another drive.

For example, the following response file would install a DB2 Administration Client on the c:\sql11ib directory, with the REBOOT and the catalog NO AUTHORIZATION options enabled

Note: The COMP keywords will be effective only if the *Install_Type* is CUSTOM.

```
FILE                = c:\sql11ib
INSTALL_TYPE       = CUSTOM
PROD               = ADMIN_CLIENT
REBOOT             = YES
DB2.CATALOG_NOAUTH = YES
```

If you specify the DB2.CATALOG_NOAUTH=YES keyword, users will not be required to have System Administrative (SYSADM) or System Controller (SYSCTRL) authority to catalog databases. This is the default setting with DB2 Client and DB2 Connect Personal Edition response files.

You should install DB2 products only on a drive which is local to the target workstation. Installing on a non-local drive can cause performance and availability problems.

Your next step is to run setup with the response file from the client workstation.

Related tasks:

- “Running setup with the response file from the client workstation on Windows” on page 133
- “Setting up shared access to a directory on Windows” on page 131

Running setup with the response file from the client workstation on Windows

Running setup with the response file from the client workstation is part of the larger task of a response file installation of DB2 on Windows.

Prerequisites:

Log on to the system with a user account that you want to use to perform the installation.

Procedure:

To perform an installation from the workstation where the DB2 products will be installed:

1. Connect to the shared directory of the network drive or CD-ROM drive by entering the following command from the command prompt:

```
net use x: \\computer_name\directory_sharename /USER:domain\username
```

where:

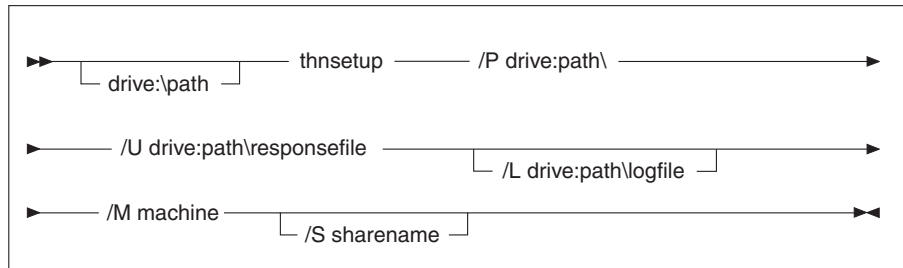
- *x*: represents the shared directory on the local drive.
- *computer_name* represents the computer name of the remote machine where the DB2 install files reside.
- *directory_sharename* represents the share name of the directory on network drive or CD-ROM drive where the DB2 install files reside.
- *domain* represents the domain where the account is defined.
- *username* represents a user that has access to this machine.

For example, to use the remote db2prods directory, which was shared as db2nt and is located on the remote server codesrv, as the local x: drive, enter the following command:

```
net use x: \\codesrv\db2nt
```

Depending on how security is set up across your network, you may have to specify the */USER* parameter.

2. Run the setup program by issuing the following from a command prompt:



where:

- /U** Specifies the fully qualified response file name. If you changed and renamed the sample response file that is provided, make sure that this parameter matches the new name. This parameter is required.
- /L** Specifies the fully qualified log file name, where setup information and any errors occurring during setup are logged. This parameter is optional.

If you do not specify the log file's name, DB2 names it db2.log. The db2.log file is located in the Start/Documents/My Documents folder.
- /I** Specifies the two-character country/region code that represents your language. If you do not specify the language, setup will determine the system language, and launch the appropriate DB2 install for that language. This parameter is optional.

For example, to install a DB2 Administration Client using a custom response file that you created called `admin.rsp` (located in the same directory as the DB2 install files), enter the following command:

```
x:\setup /U admin.rsp
```

If you are using a response file that was created using the response file generator, ensure that all the instance profiles are located in the same drive and directory as the response file that you specify.

3. Check the messages in the log file when the installation finishes.

Related tasks:

- “Creating a response file on Windows” on page 132

Installing DB2 products using Microsoft Systems Management Server (SMS)

With Microsoft Systems Management Server (SMS), you can install DB2 across a network, and set up the installation from a central location. An SMS install will minimize the amount of work the users will have to perform. This installation method is ideal if you want to roll out an installation on a large number of clients all based on the same setup.

Prerequisites:

You must have at least SMS Version 1.2 installed and configured on your network for both your SMS server and SMS workstation. Refer to *Microsoft's Systems Management Server Administrator's Guide* for your platform for more details on how to:

- Set up SMS (including setting up primary and secondary sites).
- Add clients to the SMS system.
- Set up inventory collection for clients.

Procedure:

To install DB2 products using SMS:

1. Import the DB2 install file into SMS
2. Create the SMS package on the SMS server
3. Distribute the DB2 installation package across your network

When you are using SMS, you have control over which response file you will use. You can have several different installation options, resulting in several different response files. When you configure the SMS install package, you can specify which response file to use.

Related tasks:

- “Importing the DB2 install file into SMS” on page 136
- “Creating the SMS package on the SMS server” on page 137
- “Distributing the DB2 installation package across your network” on page 138
- “Configuring db2cli.ini for a response file installation” on page 142
- “Configuring remote access to a server database” on page 140
- “Response file installation of DB2 on Windows” on page 129
- “Exporting and importing a profile” on page 142

Importing the DB2 install file into SMS

Importing the DB2 install file into SMS is part of the larger task of installing DB2 products using SMS.

To set up a package through SMS, you will use the sample SMS package description (**db2.pdf**) file and your customized response file and instance profile. If you are using a response file that was created using the response file generator, you must ensure that all the instance profiles are located in the same drive and directory as the response file that you specify.

Procedure:

To import the DB2 install files into SMS:

1. Insert the appropriate CD-ROM into the drive.
2. Start the **Microsoft SMS Administrator**. The **Microsoft SMS Administrator Logon** window opens.
3. Enter your logon ID and password, and click **OK**. The **Open SMS** window opens.
4. Select the **Packages** window type and click **OK**. The **Packages** window opens.
5. Select **File**—>**New** from the menu bar. The **Package Properties** window opens.
6. Click the **Import** push button. The **File Browser** opens. Find the **db2.pdf** file located in `x:\db2\common\`, where `x`: represents the CD-ROM drive.
7. Click **OK**.

Related tasks:

- “Creating the SMS package on the SMS server” on page 137
- “Response file installation of DB2 on Windows” on page 129

Creating the SMS package on the SMS server

Creating the SMS package on the SMS server is part of the larger task of *Installing DB2 products using SMS*.

An *SMS package* is a bundle of information that you send from the SMS server to an SMS client. The package consists of a set of commands that can be run on the client workstation. These commands could be for system maintenance, changing client configuration parameters, or installing software.

Procedure:

To create an SMS package:

1. From the **Package Properties** window, click on the **Workstations** push button. The **Setup Package For Workstations** window opens, with the imported response file and instance profile ready to use.
2. In the **Source Directory** field, enter the name of the parent directory where you put the copied DB2 files. For example, `x:\db2prods`, where `x:` represents your CD-ROM drive.
3. Select the name of the product to install from the **Workstation Command Lines** window.
4. If you changed and renamed the sample response file, click on the **Properties** push button. The **Command Line Properties** window opens. Change the value of the **Command Line** parameter to match the new response file name and path. If you are using a response file that was created using the response file generator, ensure that all the instance profiles are located in the same drive and directory as the response file that you specify.
5. Click **OK**.
6. Click the **Close** push button.
7. Click **OK** to close the opened windows. The Packages window shows the name of the new SMS package.

Related tasks:

- “Distributing the DB2 installation package across your network” on page 138
- “Importing the DB2 install file into SMS” on page 136

Distributing the DB2 installation package across your network

Distributing the DB2 installation package across your network is part of the larger task of *Installing DB2 products using SMS*.

Now that you have created the package, you have three options:

- You can distribute your SMS package and then log on locally on the client workstation to run the package. This option requires that the user account used to perform the installation belongs to the *local Administrators* group where the account is defined.
- You can distribute your SMS package and then log on remotely on the client workstation to run the package. This option requires that the user account used to perform the installation belongs to the *Domain Admins* group.
- You can set up your SMS package with an auto-install feature.

Options 1 and 2 are available to you, but for a large number of installations we recommend option 3, which will be our focus for this step.

Once sent to the client workstation, the SMS package will tell the client workstation what code to execute, and the location, on the SMS server, of that code.

Procedure:

To send the code to a client workstation:

1. Open the **Sites** window.
2. Open the **Packages** window.
3. In the **Packages** window, select the appropriate package and drag it onto the target client in the **Sites** window. The **Job Details** window opens. This window lists the package that will be sent to the client machine (Machine Path) and the command that will be executed at the workstation.
4. Select the **Run Workstation Command** check box and select the installation package that you want to use.
5. In the **Run Phase** box of the **Job Details** window, select the **Mandatory After** check box. A default mandatory date is set one week from the current date. Adjust the date as required.
6. Deselect the **Not Mandatory over Slow Link** check box. This feature is critical if you are installing across a large number of workstations. It is recommended that you stagger the installation to avoid overloading your server. For example, if you are considering an overnight install, then spread out the install time for a manageable amount of client

workstation. For more information about completing the **Job Details** window, refer to *Microsoft's Systems Management Server Administrator's Guide* for your platform.

7. When the job specifications are complete, click **OK**. You are returned to the **Job Properties** window.
8. Add a comment that explains what the job will do. For example, Install DB2 Run-Time Client.
9. Click the **Schedule** push button and the **Job Schedule** window opens. This window will arrange a priority for this job. By default, the job is low priority and all other jobs will be executed first. It is recommended that you select medium or high priority. You can also select a time to start the job.
10. Click **OK** to close the **Job Schedule** window.
11. Click **OK**.

The job is created and the package is sent to the SMS client workstation.

To run the installation on the SMS client, perform the following steps:

1. On the target SMS client workstation, log on to the workstation with a user account that belongs to the *local Administrators* group where the account is defined. This level of authority is required because a system program install is being performed instead of a user program install.
2. Start the **Package Command Manager**. The **Package Command Manager** window opens.
3. When the SMS client workstation receives the packages from the SMS server, it is listed in the **Package Name** section of the window. Select the package and click on the **Execute** push button. The installation runs automatically.
4. Following installation, you must reboot the SMS client workstation before using DB2. **Important:** If you specified REBOOT = YES in your response file, the SMS client will reboot automatically.
5. Click **Start** and select **Programs—>SMS Client—>Package Command Manager**. The **Package Command Manager** window opens.
6. Click the **Executed Commands** folder and verify the execution of the package. Similarly, you can verify completion on the SMS server by checking the status of the job and ensuring that it has been changed to complete from pending or active.

On the SMS client, open the Package Command Manager again. When the package, which you created and sent to the client, appears under the Executed Commands folder, the installation has completed.

Related tasks:

- “Creating the SMS package on the SMS server” on page 137

Configuring remote access to a server database

Once you have installed your DB2 product, you can configure your product to access remote databases individually on each client workstation using the Configuration Assistant or the command line processor. DB2 uses the **CATALOG** command to catalog remote database access information:

- The **CATALOG NODE** command specifies the protocol information on how to connect to the host or to the server.
- The **CATALOG DATABASE** command catalogs the remote database name and assigns it a local alias.
- The **CATALOG DCS** command specifies that the remote database is a host or OS/400 database. (This command is only required for DB2 Connect Personal or Enterprise Editions).
- The **CATALOG ODBC DATA SOURCE** command registers the DB2 database with the ODBC driver manager as a data source.

Prerequisites:

If you plan to roll out multiple copies of DB2 clients with identical configurations, then you can create a batch file that will run your customized script.

For example, consider the following sample batch file, `myscript.bat`, used to run the script file:

```
@echo off
cls
db2cmd catmvs.bat
```

The `DB2CMD` command initializes the DB2 environment and the `catmvs.bat` file calls the batch job of the same name.

Here is a sample catalog script file, `catmvs.bat`, that could be used to add databases to a DB2 Connect Personal Edition workstation:

```
db2 catalog tcpip node tcptst1 remote mvshost server 446
db2 catalog database mvbdb at node tcptst1 authentication dcs
db2 catalog dcs database mvbdb as mvs_locator
db2 catalog system odbc data source mvbdb
db2 terminate
exit
```

Procedure:

You can either send these files to your client workstations manually or use SMS and have the script execute automatically after the installation and reboot have completed. To create another SMS package with the catalog script, perform the following steps:

1. Start the **SMS Administrator**. The **Open SMS** window opens.
2. Select the **Packages** window type and click **OK**. The **Packages** window opens.
3. Select **File**—>**New** from the menu bar. The **Package Properties** window opens.
4. Enter a name for your new package. For example, batchpack.
5. Enter a comment about the package. For example, Package for batch file.
6. Click on the **Workstations** push button. The **Setup Package for Workstations** window opens.
7. Enter the source directory. Ensure that the source directory is a location that both the server and the client have access to, and that contains the batch file that is to be run from the client workstation.
8. Under the **Workstation Command Lines** section, click on **New**. The **Command Line Properties** window opens.
9. Enter a command name.
10. Enter the command line.
11. Click the check box for the platforms that should be supported, under the **Supported Platforms** section.
12. Click **OK**.
13. Click **Close**.
14. Click **OK**.

Distribute this package in the same way as an installation package.

Related tasks:

- “Configuring db2cli.ini for a response file installation” on page 142
- “Installing DB2 products using Microsoft Systems Management Server (SMS)” on page 135
- “Distributing the DB2 installation package across your network” on page 138

Configuring db2cli.ini for a response file installation

The `db2cli.ini` file is an ASCII file which initializes the DB2 CLI configuration. This file is shipped to help you get started and can be found in the `x:\sqllib` directory, where `x:\sqllib` represents the install path for DB2.

Procedure:

If you need to use any specific CLI optimization values or CLI parameters, you can use your customized `db2cli.ini` file for your DB2 client workstations. To do so, copy your `db2cli.ini` file to the DB2 install directory (e.g. `c:\Program Files\IBM\SQLLIB`) on each DB2 client workstation.

Related tasks:

- “Configuring remote access to a server database” on page 140
- “Installing DB2 products using Microsoft Systems Management Server (SMS)” on page 135

Exporting and importing a profile

Procedure:

If you did not use a configuration profile when you installed your DB2 product using the response file that was created by the response file generator, you can enter the `db2cfexp` command to create a configuration profile. The `db2cfimp` command can then be used to import a configuration profile.

You can also use the CA to export and import a configuration profile.

Chapter 5. Migrating a Satellite Environment

Migrating Servers on Windows	143	Migration recommendations	158
Migrating the Satellite Control Server		Backing up databases before DB2	
(Windows)	143	migration	160
Migration restrictions	144	Space considerations for DB2 migration	161
Migration recommendations	146	Recording system configuration settings	
Backing up databases before DB2		before DB2 migration	162
migration	147	Verifying that your databases are ready	
Space considerations for DB2 migration	148	for migration	162
Recording system configuration settings		Taking a V6 or V7 DB2 server offline for	
before DB2 migration	149	DB2 migration	164
Changing the diagnostic error level before		Migrating instances (UNIX)	165
DB2 migration	150	Migrating the DB2 Administration Server	
Verifying that your databases are ready		(DAS)	167
for migration	151	Migrating databases	167
Taking a V6 or V7 DB2 server offline for		Migrating DB2 Personal Edition	169
DB2 migration	152	Migrating DB2 Personal Edition	
Migrating databases	153	(Windows)	169
Migrating Servers on UNIX	155	Preparing to migrate DB2 Personal	
Migrating the Satellite Control Server		Edition (Windows)	170
(UNIX)	155	Migrating databases on DB2 Personal	
Migration restrictions	156	Edition (Windows)	173

Migrating Servers on Windows

The sections that follow describe how to migrate DB2 servers on Windows-based platforms.

Migrating the Satellite Control Server (Windows)

This topic lists the steps that you must perform to migrate a Windows-based satellite control server to DB2 Enterprise Server Edition, Version 8. Migration is required if you have pre-DB2 Version 8 instances and databases you would like to continue using with DB2 Version 8.

Prerequisites:

Ensure that you review:

- Migration recommendations
- Migration restrictions
- Space considerations for DB2 migration

Procedure:

To migrate the satellite control server for Windows:

1. Record configuration settings before DB2 migration.
2. Change the diagnostic error level.
3. Take the DB2 server offline for DB2 migration.
4. Verify that databases are ready for DB2 migration.
5. Back up your databases.
6. Install your DB2 server:
 - Enterprise Server Edition (partitioned)
7. Migrate databases.
8. *Optional*: Migrate DB2 Explain tables.

Related tasks:

- “Recording system configuration settings before DB2 migration” on page 149
- “Changing the diagnostic error level before DB2 migration” on page 150
- “Taking a V6 or V7 DB2 server offline for DB2 migration” on page 152
- “Verifying that your databases are ready for migration” on page 151
- “Backing up databases before DB2 migration” on page 147
- “Installing a partitioned DB2 server (Windows)” on page 29
- “Migrating databases” on page 153
- “Migrating Explain tables” in the *Quick Beginnings for DB2 Servers*

Migration restrictions

You should be aware of the following restrictions before you migrate to DB2 Version 8:

- Migration is only supported from:
 - DB2 Version 6.x or Version 7.x. (All platforms supported in Version 6.x and Version 7.x; Linux must be at Version 6 FixPak 2.)
 - DB2 DataJoiner V2.1.1 32-bit (AIX, Windows NT, and Solaris Operating Environment).
- Issuing the **migrate database** command from a DB2 Version 8 client to migrate a database to a DB2 Version 8 server is supported; however, issuing the migration command from an DB2 Version 6 or Version 7 client to migrate a database to a DB2 Version 8 server is not supported.
- When migrating from DB2 DataJoiner V2.1.1, DB2 Relational Connect is required to support non-IBM data sources.
- Migration between platforms is not supported. For example, you cannot migrate a database from a DB2 server on Windows to a DB2 server on UNIX.

- Migrating a partitioned database system that has multiple computers requires that database migration be performed after DB2 Version 8 is installed on all participating computers. Any DB2 migration commands need to be run on each of the participating computers.
- Windows allows only one version of DB2 to be installed on a computer. For example, if you have DB2 Version 7 and install DB2 Version 8, DB2 Version 7 will be removed during the installation. All instances are migrated during DB2 installation on Windows operating systems.
- User objects within your database cannot have DB2 Version 8 reserved schema names as object qualifiers. These reserved schema names include: SYSCAT, SYSSTAT, and SYSFUN.
- User-defined distinct types using the names BIGINT, REAL, DATALINK, or REFERENCE must be renamed before migrating the database.
- You cannot migrate a database that is in one of the following states:
 - Backup pending
 - Roll-forward pending
 - One or more table spaces not in a normal state
 - Transaction inconsistent
- Restoration of down-level (DB2 Version 6.x or Version 7.x) database backups is supported, but the rolling forward of down-level logs is not supported.
- Database transactions executed between database backup time and the time DB2 Version 8 migration is completed are not recoverable.
- To migrate a DB2 Version 7 AIX Version 4 64-bit instance:
 - Upgrade your AIX operating system to AIX Version 5:
 1. Upgrade your operating system to AIX Version 5.
 2. Upgrade DB2 Version 7 with DB2 Version 7 FixPak 4 for AIX 5.
 3. Update your instances using the `/usr/lpp/db2_07_01/instance/db2iupdt` command.
 4. Ensure your database continues to work. It is not recommended to proceed directly to the next step without confirming your database works in AIX Version 5 on DB2 Version 7.
 5. Install DB2 Version 8 for AIX Version 5
 6. Migrate your instances using the `/usr/opt/db2_08_01/instance/db2imigr` command.
 - Remain with AIX Version 4:
 1. Drop your instances.
 2. Recreate them as 32-bit instances. You may have to reconfigure your instance parameters.
 3. Install DB2 Version 8 for AIX Version 4.

4. Migrate your instances using the `/usr/opt/db2_08_01/instance/db2imigr` command.

Related concepts:

- “Migration recommendations” on page 146

Related tasks:

- “Migrating DB2 (Windows)” in the *Quick Beginnings for DB2 Servers*
- “Migrating DB2 (UNIX)” in the *Quick Beginnings for DB2 Servers*

Migration recommendations

Consider the following recommendations when planning your database migration:

Perform hardware and operating system upgrades separately from DB2[®] migration

Performing hardware and operating system upgrades separately from DB2 migration allows for easier problem determination should you encounter migration difficulties. If you upgrade your software or hardware prior to migrating to DB2, ensure that your system is operating acceptably before attempting DB2 migration.

Down level server support

As you move your environment from DB2 Version 7 to DB2 Version 8, if you are in a situation where you migrate your DB2 clients to Version 8 before you migrate all of your DB2 servers to Version 8, there are several restrictions and limitations. These restrictions and limitations are not associated with DB2 Connect; nor with zSeries, OS/390, or iSeries database servers. To avoid the known restrictions and limitations, you should migrate all of your DB2 servers to Version 8 before you migrate any of your DB2 clients to Version 8.

Benchmark DB2 performance

Run a number of test queries before migrating DB2. Record the exact environment conditions when queries are run. Also, keep a record of the `db2expln` command output for each test query. Compare the results before and after migration. This practice may help to identify and correct any performance degradation.

Devise a plan to back out of a migration

There is no utility to reverse a migration. If you must back out of a migration, you might have to remove DB2 Version 8 code from your system, reinstall the previous version of DB2 to recreate back-level instances, and restore your database backups. Should you have to back out of a migration, current database backups and a detailed record of database and database configuration settings are essential.

Migrate to DB2 Version 8 in a test environment

Migrate to DB2 Version 8 in a test environment before migrating your production environment. This practise will allow you to address migration difficulties and make sure applications and tools work properly before committing your production environment to the migration process.

Migrating Instances with DB2 DataPropagator Replication

Before migrating an instance of DataJoiner, DB2 for UNIX, or DB2 for Windows where you are running the Capture or Apply programs for DB2 DataPropagator, be sure to read the migration documentation for DB2 DataPropagator Version 8. You must prepare to migrate your replication environment before you migrate the DB2 or DataJoiner instance. You must also perform steps immediately after the migration of your DB2 or DataJoiner instance. Migration documentation for DB2 DataPropagator Version 8 can be found at the <http://www.ibm.com/software/data/dpropr/library.html> Web site.

Related concepts:

- “Benchmark testing” in the *Administration Guide: Performance*
- “Explain tools” in the *Administration Guide: Performance*

Related tasks:

- “Migrating DB2 (Windows)” in the *Quick Beginnings for DB2 Servers*
- “Migrating DB2 (UNIX)” in the *Quick Beginnings for DB2 Servers*

Related reference:

- “db2expln - DB2 SQL Explain Tool Command” in the *Command Reference*
- “DB2 Universal Database planned incompatibilities” in the *Administration Guide: Planning*
- “Version 8 incompatibilities between releases” in the *Administration Guide: Planning*
- “Version 7 incompatibilities between releases” in the *Administration Guide: Planning*

Backing up databases before DB2 migration

Before you start the migration process it is recommended that you perform an *offline* backup of your databases. If an error should occur during the migration process, database backups are required for recovery.

This topic does not provide the complete syntax for the backup command. For the complete syntax, refer to the Related reference at the end of this topic.

Prerequisites:

- To backup a database, you require SYSADM, SYSCTRL, or SYSMOINT authority.
- Databases must be cataloged. To view a list of all the cataloged databases in the current instance, enter the following command:

```
db2 list database directory
```

Procedure:

Back up each of your local databases using the backup database command:

BACKUP Command



where:

DATABASE *database-alias*

Specifies the alias of the database to back up.

USER *username*

Identifies the user name under which to back up the database.

USING *password*

Is the password used to authenticate the user name. If the password is omitted, the user is prompted to enter it.

Related concepts:

- “System administration authority (SYSADM)” in the *Administration Guide: Implementation*

Related reference:

- “BACKUP DATABASE Command” in the *Command Reference*
- “Space considerations for DB2 migration” on page 148

Space considerations for DB2 migration

This topic provides information about space requirements for DB2 migration.

Table spaces

Ensure that you have sufficient table space for databases you are migrating. System catalog table space is required for both old and new database catalogs during migration. The amount of space required will vary, depending on the complexity of the database, as well as the number and size of database objects. General recommendations:

Table 11. Catalog table space recommendations

Table space	Recommended space
system catalog table space (SYSCATSPACE)	2 x the space currently occupied
temporary table space (TEMPSPACE1 is the default name)	2 x the system catalog table space

To check the size of your table spaces you can use the following commands:

```
db2 list database directory
db2 connect to database_alias
db2 list tablespaces show detail
```

For the system catalog table space, free pages should be equal to or greater than used pages. Total pages for the temporary table space should be twice the amount of total pages for the system catalog table space. To increase the amount of space to a DMS tablespace, you can add additional containers.

Log file space

You should consider increasing (doubling) the values of *logfilsiz*, *logprimary*, and *logsecond* to prevent log file space from running out. If your system catalog table space is an SMS type of table space, you should consider updating the database configuration parameters that are associated with the log files.

Related tasks:

- “Adding a container to a DMS table space” in the *Administration Guide: Implementation*
- “Migrating DB2 (Windows)” in the *Quick Beginnings for DB2 Servers*
- “Migrating DB2 (UNIX)” in the *Quick Beginnings for DB2 Servers*

Recording system configuration settings before DB2 migration

It is recommended that you record database and database manager configuration settings before DB2 migration. Configuration records can be used to verify that migration was successful and may also be useful in problem determination, should you encounter post-migration difficulties.

After you have migrated DB2, it is recommended that you compare these records with post-migration settings to ensure that existing settings were migrated successfully.

This topic lists several database commands. For references to complete command syntax, refer to the related reference section at the end of this topic.

Procedure:

1. Save a copy of your database configuration settings. Configuration parameters for a database should be the same on each computer in a partitioned database system. If not, save a copy of the database configuration settings for each partition. You can compare configuration settings before and after migration to ensure that they have been migrated properly. You can retrieve database configuration settings using the **db2 get database configuration for *database_alias*** command. Perform this task for each database you are migrating.
2. Save a copy of your database manager configuration settings. You can retrieve database manager configuration settings using the **db2 get database manager configuration** command.
3. Save a record of the tablespaces for each database you are migrating. You can retrieve a list of tablespaces using the **db2 list tablespaces** command.
4. Save a list of packages for each database you are migrating. You can retrieve a list of packages using the **db2 list packages** command.

Related reference:

- “GET DATABASE CONFIGURATION Command” in the *Command Reference*
- “GET DATABASE MANAGER CONFIGURATION Command” in the *Command Reference*
- “LIST PACKAGES/TABLES Command” in the *Command Reference*
- “LIST TABLESPACES Command” in the *Command Reference*

Changing the diagnostic error level before DB2 migration

For the duration of migration activities, change the diagnostic error level to 4. The diagnostic error level 4 records all errors, warnings and informational messages. This information can be used for problem determination should you encounter migration errors. The `diagpath` configuration parameter is used to specify the directory that contains the error file, event log file (on Windows only), alert log file, and any dump files that may be generated based on the value of the `diaglevel` parameter.

Procedure:

The diagnostic error level can be set in the database manager configuration file using the following command:

```
db2 update dbm configuration using diaglevel 4
```

The `diagpath` parameter can be set in the database manager configuration file using the following command:

```
db2 update dbm configuration using diagpath directory
```

where *directory* is the location you have chosen to store your log files.

Verifying that your databases are ready for migration

This task describes how to use the **db2ckmig** command to verify that your databases are ready for migration.

Prerequisites:

Ensure that the `migration.log`, found in the instance owner's home directory, contains the following text: Version of DB2CKMIG being run: VERSION 8.

Procedure:

Enter the **db2ckmig** command to verify that databases owned by the current instance are ready to be migrated. The **db2ckmig** command ensures that:

- A database is not in a inconsistent state
- A database is not in backup pending state
- A database is not in rollforward pending state
- Tablespaces are in a normal state

DB2CKMIG command

```
▶—db2ckmig—[database_alias—/l—drive:\path\filename—]—▶  
          [/e—]—  
▶—[/u—userid—/p—password—]—▶
```

where:

database_alias

Specifies a *database_alias* name of a database to be verified for migration. This parameter is required if the `/e` parameter is not specified.

`/e`

Specifies that all cataloged databases are to be verified for migration. This parameter is required if the *database_alias* parameter is not specified.

`/l drive:\path\filename`

Specifies a drive, target path and filename to keep a list of errors and warnings generated for the scanned database. The path variable is

optional; if you do not specify a path, the path from which you execute the **db2ckmig** command will be used. You must specify a filename.

/u userid

Specifies the user account used to connect to the database. This parameter must be specified if you are logged on as a user without connect authority.

/p password

Specifies the password of the user account used to connect to the database.

The **db2ckmig** command is located in the \db2\Windows\utilities directory on your DB2 version 8 product CD-ROM.

Related tasks:

- “Migrating DB2 (UNIX)” in the *Quick Beginnings for DB2 Servers*

Taking a V6 or V7 DB2 server offline for DB2 migration

This task describes how to take your DB2 Version 6 or Version 7 server offline for DB2 migration. Before you can continue with the migration process, you must stop the DB2 license service, stop all command line processor sessions, disconnect applications and users, and stop the database manager.

Prerequisites:

Ensure that:

- Your system meets the installation requirements for DB2 Version 8 before starting the migration process.
- You have SYSADM authority.

Procedure:

To take your system offline:

1. Stop the DB2 license service by entering the **db2licd -end** command.
2. On Windows 2000, the properties of a service can be set so that it restarts if the service fails. If the *restart on failure* option is set for any DB2 services, it must be disabled before proceeding.
3. Stop all command line processor sessions by entering the **db2 terminate** command in each session that was running the command line processor.
4. Disconnect all applications and users. To get a list of all database connections for the current instance, enter the **db2 list applications** command. If all applications are disconnected, this command will return the following message:

SQL1611W No data was returned by the Database System Monitor.
SQLSTATE=00000

You can disconnect applications and users by issuing the **db2 force applications** command.

5. When all applications and users are disconnected, stop each database manager instance by entering the **db2stop** command.

Related reference:

- “db2stop - Stop DB2 Command” in the *Command Reference*
- “FORCE APPLICATION Command” in the *Command Reference*
- “LIST APPLICATIONS Command” in the *Command Reference*
- “Installation requirements for DB2 servers (Windows)” on page 17
- “Installation requirements for partitioned DB2 servers (AIX)” on page 69
- “Installation requirements for partitioned DB2 servers (HP-UX)” in the *Quick Beginnings for DB2 Servers*
- “Installation requirements for partitioned DB2 servers (Solaris Operating Environment)” in the *Quick Beginnings for DB2 Servers*
- “Installation requirements for partitioned DB2 servers (Linux)” in the *Quick Beginnings for DB2 Servers*
- “Installation requirements for DB2 servers (AIX)” on page 55
- “Installation requirements for DB2 servers (HP-UX)” in the *Quick Beginnings for DB2 Servers*
- “Installation requirements for DB2 servers (Linux)” in the *Quick Beginnings for DB2 Servers*
- “Installation requirements for DB2 servers (Solaris)” in the *Quick Beginnings for DB2 Servers*
- “Installation requirements for a partitioned DB2 server (Windows)” on page 33

Migrating databases

This task is part of the main task of *Migrating DB2*.

Prerequisites:

You require SYSADM authority.

Restrictions:

Migration is only supported from:

- DB2 Version 6.x or Version 7.x. (all platforms supported in Version 6.x and Version 7.x)

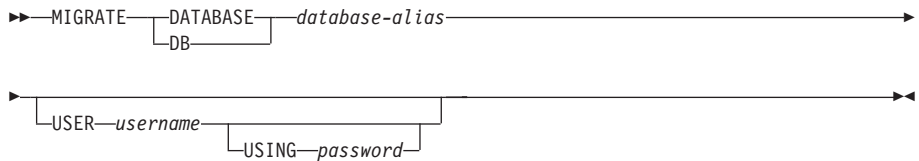
- DB2 DataJoiner V2.1.1 (AIX, Windows NT, and Solaris Operating Environment).

Procedure:

To migrate a DB2 database:

1. Migrate the database using the **db2 migrate database** command.

DB2 MIGRATE DATABASE command



where:

DATABASE *database-alias*

Specifies the alias of the database to be migrated to the currently installed version of the database manager.

USER *username*

Identifies the user name under which the database is to be migrated.

USING *password*

The password used to authenticate the user name. If the password is omitted, but a user name was specified, the user is prompted to enter it.

2. *Optional:* Update statistics. When database migration is complete, old statistics that are used to optimize query performance are retained in the catalogs. However, DB2 Version 8 has statistics that are modified or do not exist in DB2 Version 6 or DB2 Version 7. To take advantage of these statistics, you may want to execute the **runstats** command on tables, particularly those tables that are critical to the performance of your SQL queries.
3. *Optional:* Rebind packages. During database migration, all existing packages are invalidated. After the migration process, each package is rebuilt when it is used for the first time by the DB2 Version 8 database manager. You can run the **db2rbind** command to rebuild all packages stored in the database.
4. *Optional:* Revoke EXECUTE privileges on external stored procedures that contain SQL data access from PUBLIC. During database migration, EXECUTE privileges are granted to PUBLIC for all existing functions, methods, and external stored procedures. This will cause a security exposure for external stored procedures that contain SQL data access

which allow users to access SQL objects for which they would not otherwise have privileges. Revoke the privileges by entering the **db2undgp - r** command.

5. *Optional*: Migrate DB2 Explain tables
6. *Optional*: If you recorded configuration settings before migration, you might want to compare pre-migration configuration settings to current configuration settings to verify successful migration. Verify:
 - database configuration parameter settings
 - database manger configuration parameter settings
 - tablespaces records
 - packages records

Note: During migration, the database configuration parameter *maxappls* is set to automatic. If you want it set to a different value, you should update it manually.

Related tasks:

- “Recording system configuration settings before DB2 migration” on page 149
- “Migrating Explain tables” in the *Quick Beginnings for DB2 Servers*

Related reference:

- “MIGRATE DATABASE Command” in the *Command Reference*
- “LIST DATABASE DIRECTORY Command” in the *Command Reference*
- “db2rbind - Rebind all Packages Command” in the *Command Reference*

Migrating Servers on UNIX

The sections that follow describe how to migrate DB2 servers on UNIX-based platforms.

Migrating the Satellite Control Server (UNIX)

This topic lists the steps for migrating to DB2 Version 8 Satellite Control Server on UNIX.

Migration is required if you have pre-DB2 Version 8 instances and databases you would like to use with DB2 Version 8.

Prerequisites:

Ensure that you review:

- Migration recommendations

- Migration restrictions
- Space considerations for DB2 migration

Procedure:

To migrate the satellite control server for UNIX:

1. Record configuration settings before DB2 migration.
2. Change the diagnostic error level.
3. Take the DB2 server offline for DB2 migration.
4. Back up your databases.
5. *Optional:* If you will be using replication, you must archive all of the DB2 log files.
6. Install your DB2 server:
 - Enterprise Server Edition (partitioned):
 - AIX
7. Migrate instances.
8. *Optional:* If you have created a DB2 tools catalog and want to use your existing pre-Version 8 scripts and schedules (for the Control Center), you must migrate the DB2 Administration Server.
9. Migrate databases.

Related tasks:

- “Recording system configuration settings before DB2 migration” on page 149
- “Changing the diagnostic error level before DB2 migration” on page 150
- “Taking a V6 or V7 DB2 server offline for DB2 migration” on page 152
- “Backing up databases before DB2 migration” on page 147
- “Installing a partitioned DB2 server (AIX)” on page 67
- “Migrating instances (UNIX)” on page 165
- “Migrating the DB2 Administration Server (DAS)” on page 167
- “Migrating databases” on page 153

Migration restrictions

You should be aware of the following restrictions before you migrate to DB2 Version 8:

- Migration is only supported from:
 - DB2 Version 6.x or Version 7.x. (All platforms supported in Version 6.x and Version 7.x; Linux must be at Version 6 FixPak 2.)
 - DB2 DataJoiner V2.1.1 32-bit (AIX, Windows NT, and Solaris Operating Environment).

- Issuing the **migrate database** command from a DB2 Version 8 client to migrate a database to a DB2 Version 8 server is supported; however, issuing the migration command from an DB2 Version 6 or Version 7 client to migrate a database to a DB2 Version 8 server is not supported.
- When migrating from DB2 DataJoiner V2.1.1, DB2 Relational Connect is required to support non-IBM data sources.
- Migration between platforms is not supported. For example, you cannot migrate a database from a DB2 server on Windows to a DB2 server on UNIX.
- Migrating a partitioned database system that has multiple computers requires that database migration be performed after DB2 Version 8 is installed on all participating computers.
- Windows allows only one version of DB2 to be installed on a computer. For example, if you have DB2 Version 7 and install DB2 Version 8, DB2 Version 7 will be removed during the installation. All instances are migrated during DB2 installation on Windows operating systems.
- User objects within your database cannot have DB2 Version 8 reserved schema names as object qualifiers. These reserved schema names include: SYSCAT, SYSSTAT, and SYSFUN.
- User-defined distinct types using the names BIGINT, REAL, DATALINK, or REFERENCE must be renamed before migrating the database.
- You cannot migrate a database that is in one of the following states:
 - Backup pending
 - Roll-forward pending
 - One or more table spaces not in a normal state
 - Transaction inconsistent
- Restoration of down-level (DB2 Version 6.x or Version 7.x) database backups is supported, but the rolling forward of down-level logs is not supported.
- Database transactions executed between database backup time and the time DB2 Version 8 migration is completed are not recoverable.
- To migrate a DB2 Version 7 AIX Version 4 64-bit instance:
 - Upgrade your AIX operating system to AIX Version 5:
 1. Upgrade your operating system to AIX Version 5.
 2. Upgrade DB2 Version 7 with DB2 Version 7 FixPak 4 for AIX 5.
 3. Update your instances using the `/usr/lpp/db2_07_01/instance/db2iupdt` command.
 4. Ensure your database continues to work. It is not recommended to proceed directly to the next step without confirming your database works in AIX Version 5 on DB2 Version 7.
 5. Install DB2 Version 8 for AIX Version 5

6. Migrate your instances using the `/usr/opt/db2_08_01/instance/db2imigr` command.
- Remain with AIX Version 4:
 1. Drop your instances.
 2. Recreate them as 32-bit instances. You may have to reconfigure your instance parameters.
 3. Install DB2 Version 8 for AIX Version 4.
 4. Migrate your instances using the `/usr/opt/db2_08_01/instance/db2imigr` command.

Related concepts:

- “Migration recommendations” on page 146

Related tasks:

- “Migrating DB2 (Windows)” in the *Quick Beginnings for DB2 Servers*
- “Migrating DB2 (UNIX)” in the *Quick Beginnings for DB2 Servers*

Migration recommendations

Consider the following recommendations when planning your database migration:

Perform hardware and operating system upgrades separately from DB2[®] migration

Performing hardware and operating system upgrades separately from DB2 migration allows for easier problem determination should you encounter migration difficulties. If you upgrade your software or hardware prior to migrating to DB2, ensure that your system is operating acceptably before attempting DB2 migration.

Down level server support

As you move your environment from DB2 Version 7 to DB2 Version 8, if you are in a situation where you migrate your DB2 clients to Version 8 before you migrate all of your DB2 servers to Version 8, there are several restrictions and limitations. These restrictions and limitations are not associated with DB2 Connect; nor with zSeries, OS/390, or iSeries database servers. To avoid the known restrictions and limitations, you should migrate all of your DB2 servers to Version 8 before you migrate any of your DB2 clients to Version 8.

Benchmark DB2 performance

Run a number of test queries before migrating DB2. Record the exact environment conditions when queries are run. Also, keep a record of the `db2expln` command output for each test query. Compare the results before and after migration. This practice may help to identify and correct any performance degradation.

Devise a plan to back out of a migration

There is no utility to reverse a migration. If you must back out of a migration, you might have to remove DB2 Version 8 code from your system, reinstall the previous version of DB2 to recreate back-level instances, and restore your database backups. Should you have to back out of a migration, current database backups and a detailed record of database and database configuration settings are essential.

Migrate to DB2 Version 8 in a test environment

Migrate to DB2 Version 8 in a test environment before migrating your production environment. This practise will allow you to address migration difficulties and make sure applications and tools work properly before committing your production environment to the migration process.

Migrating DataJoiner instances

Before migrating an instance of DataJoiner, DB2 for UNIX, or DB2 for Windows where you are running the Capture or Apply programs for DB2 DataPropagator, be sure to read the migration documentation for DB2 DataPropagator Version 8. You must prepare to migrate your replication environment before you migrate the DB2 or DataJoiner instance. You must also perform steps immediately after the migration of your DB2 or DataJoiner instance. Migration documentation for DB2 DataPropagator Version 8 can be found at the <http://www.ibm.com/software/data/dprop/library.html> Web site.

Related concepts:

- “Benchmark testing” in the *Administration Guide: Performance*
- “Explain tools” in the *Administration Guide: Performance*

Related tasks:

- “Migrating DB2 (Windows)” in the *Quick Beginnings for DB2 Servers*
- “Migrating DB2 (UNIX)” in the *Quick Beginnings for DB2 Servers*

Related reference:

- “db2expln - DB2 SQL Explain Tool Command” in the *Command Reference*
- “DB2 Universal Database planned incompatibilities” in the *Administration Guide: Planning*
- “Version 8 incompatibilities between releases” in the *Administration Guide: Planning*
- “Version 7 incompatibilities between releases” in the *Administration Guide: Planning*

Backing up databases before DB2 migration

Before you start the migration process it is recommended that you perform an *offline* backup of your databases. If an error should occur during the migration process, database backups are required for recovery.

This topic does not provide the complete syntax for the backup command. For the complete syntax, refer to the Related reference at the end of this topic.

Prerequisites:

- To backup a database, you require SYSADM, SYSCTRL, or SYSMAINT authority.
- Databases must be cataloged. To view a list of all the cataloged databases in the current instance, enter the following command:

```
db2 list database directory
```

Procedure:

Back up each of your local databases using the backup database command:

BACKUP Command

```
▶▶—BACKUP—┬──DATABASE──┬──database-alias──┬──────────────────────────────────┬──▶▶  
             └──DB──┬──┴──┬──USER—username──┬──┬──USING—password──┬──┴──
```

where:

DATABASE *database-alias*

Specifies the alias of the database to back up.

USER *username*

Identifies the user name under which to back up the database.

USING *password*

Is the password used to authenticate the user name. If the password is omitted, the user is prompted to enter it.

Related concepts:

- “System administration authority (SYSADM)” in the *Administration Guide: Implementation*

Related reference:

- “BACKUP DATABASE Command” in the *Command Reference*
- “Space considerations for DB2 migration” on page 148

Space considerations for DB2 migration

This topic provides information about space requirements for DB2 migration.

Table spaces

Ensure that you have sufficient table space for databases you are migrating. System catalog table space is required for both old and new database catalogs during migration. The amount of space required will vary, depending on the complexity of the database, as well as the number and size of database objects. General recommendations:

Table 12. Catalog table space recommendations

Table space	Recommended space
system catalog table space (SYSCATSPACE)	2 x the space currently occupied
temporary table space (TEMPSPACE1 is the default name)	2 x the system catalog table space

To check the size of your table spaces you can use the following commands:

```
db2 list database directory
db2 connect to database_alias
db2 list tablespaces show detail
```

For the system catalog table space, free pages should be equal to or greater than used pages. Total pages for the temporary table space should be twice the amount of total pages for the system catalog table space. To increase the amount of space to a DMS tablespace, you can add additional containers.

Log file space

You should consider increasing (doubling) the values of *logfilsiz*, *logprimary*, and *logsecond* to prevent log file space from running out. If your system catalog table space is an SMS type of table space, you should consider updating the database configuration parameters that are associated with the log files.

Related tasks:

- “Adding a container to a DMS table space” in the *Administration Guide: Implementation*
- “Migrating DB2 (Windows)” in the *Quick Beginnings for DB2 Servers*
- “Migrating DB2 (UNIX)” in the *Quick Beginnings for DB2 Servers*

Recording system configuration settings before DB2 migration

It is recommended that you record database and database manager configuration settings before DB2 migration. Configuration records can be used to verify that migration was successful and may also be useful in problem determination, should you encounter post-migration difficulties.

After you have migrated DB2, it is recommended that you compare these records with post-migration settings to ensure that existing settings were migrated successfully.

This topic lists several database commands. For references to complete command syntax, refer to the related reference section at the end of this topic.

Procedure:

1. Save a copy of your database configuration settings. Configuration parameters for a database should be the same on each computer in a partitioned database system. If not, save a copy of the database configuration settings for each partition. You can compare configuration settings before and after migration to ensure that they have been migrated properly. You can retrieve database configuration settings using the **db2 get database configuration for *database_alias*** command. Perform this task for each database you are migrating.
2. Save a copy of your database manager configuration settings. You can retrieve database manager configuration settings using the **db2 get database manager configuration** command.
3. Save a record of the tablespaces for each database you are migrating. You can retrieve a list of tablespaces using the **db2 list tablespaces** command.
4. Save a list of packages for each database you are migrating. You can retrieve a list of packages using the **db2 list packages** command.

Related reference:

- “GET DATABASE CONFIGURATION Command” in the *Command Reference*
- “GET DATABASE MANAGER CONFIGURATION Command” in the *Command Reference*
- “LIST PACKAGES/TABLES Command” in the *Command Reference*
- “LIST TABLESPACES Command” in the *Command Reference*

Verifying that your databases are ready for migration

This task describes how to use the **db2ckmig** command to verify that your databases are ready for migration.

Prerequisites:

Ensure that the migration.log, found in the instance owner's home directory, contains the following text: Version of DB2CKMIG being run: VERSION 8.

Procedure:

Enter the **db2ckmig** command to verify that databases owned by the current instance are ready to be migrated. The **db2ckmig** command ensures that:

- A database is not in a inconsistent state
- A database is not in backup pending state
- A database is not in rollforward pending state
- Tablespaces are in a normal state

DB2CKMIG command

```
▶▶ db2ckmig database_alias /l drive:\path\filename
      |_____|
      /e
      |_____|
      /u userid /p password
```

where:

database_alias

Specifies a *database_alias* name of a database to be verified for migration. This parameter is required if the /e parameter is not specified.

/e

Specifies that all cataloged databases are to be verified for migration. This parameter is required if the *database_alias* parameter is not specified.

/l *drive:\path\filename*

Specifies a drive, target path and filename to keep a list of errors and warnings generated for the scanned database. The path variable is optional; if you do not specify a path, the path from which you execute the **db2ckmig** command will be used. You must specify a filename.

/u *userid*

Specifies the user account used to connect to the database. This parameter must be specified if you are logged on as a user without connect authority.

/p *password*

Specifies the password of the user account used to connect to the database.

The **db2ckmig** command is located in the `\db2\Windows\utilities` directory on your DB2 version 8 product CD-ROM.

Related tasks:

- “Migrating DB2 (UNIX)” in the *Quick Beginnings for DB2 Servers*

Taking a V6 or V7 DB2 server offline for DB2 migration

This task describes how to take your DB2 Version 6 or Version 7 server offline for DB2 migration. Before you can continue with the migration process, you must stop the DB2 license service, stop all command line processor sessions, disconnect applications and users, and stop the database manager.

Prerequisites:

Ensure that:

- Your system meets the installation requirements for DB2 Version 8 before starting the migration process.
- You have SYSADM authority.

Procedure:

To take your system offline:

1. Stop the DB2 license service by entering the **db2licd -end** command.
2. On Windows 2000, the properties of a service can be set so that it restarts if the service fails. If the *restart on failure* option is set for any DB2 services, it must be disabled before proceeding.
3. Stop all command line processor sessions by entering the **db2 terminate** command in each session that was running the command line processor.
4. Disconnect all applications and users. To get a list of all database connections for the current instance, enter the **db2 list applications** command. If all applications are disconnected, this command will return the following message:

```
SQL1611W No data was returned by the Database System Monitor.  
SQLSTATE=00000
```

You can disconnect applications and users by issuing the **db2 force applications** command.

5. When all applications and users are disconnected, stop each database manager instance by entering the **db2stop** command.

Related reference:

- “db2stop - Stop DB2 Command” in the *Command Reference*
- “FORCE APPLICATION Command” in the *Command Reference*

- “LIST APPLICATIONS Command” in the *Command Reference*
- “Installation requirements for DB2 servers (Windows)” on page 17
- “Installation requirements for partitioned DB2 servers (AIX)” on page 69
- “Installation requirements for partitioned DB2 servers (HP-UX)” in the *Quick Beginnings for DB2 Servers*
- “Installation requirements for partitioned DB2 servers (Solaris Operating Environment)” in the *Quick Beginnings for DB2 Servers*
- “Installation requirements for partitioned DB2 servers (Linux)” in the *Quick Beginnings for DB2 Servers*
- “Installation requirements for DB2 servers (AIX)” on page 55
- “Installation requirements for DB2 servers (HP-UX)” in the *Quick Beginnings for DB2 Servers*
- “Installation requirements for DB2 servers (Linux)” in the *Quick Beginnings for DB2 Servers*
- “Installation requirements for DB2 servers (Solaris)” in the *Quick Beginnings for DB2 Servers*
- “Installation requirements for a partitioned DB2 server (Windows)” on page 33

Migrating instances (UNIX)

This task is part of the main task of *Migrating DB2 (UNIX)*.

You can migrate existing DB2 Version 6 or DB2 Version 7 instances using the **db2imigr** command. Migrating instances is done after installing DB2 Version 8.

The **db2imigr** command does the following:

- Checks cataloged databases owned by the instance to make sure they are ready for migration.
- Runs the **db2icrt** command to create the DB2 Version 8 instance.
- Updates system and local database directories to a Version 8 format.
- Merges the pre-DB2 Version 8 database manager configuration with DB2 Version 8 database manager configuration.

Prerequisites:

You must be logged in as a user with root authority.

Before running the **db2imigr** command, it is recommended:

- That /tmp have up to 70 percent free space. The instance migration trace file is written to /tmp.

- That you run the **db2ckmig** command manually and resolve any problems prior to running the **db2imigr**. The **db2imigr** command will not migrate as long as the **db2ckmig** command finds problems.

Restrictions:

Migration is only supported from:

- DB2 Version 6.x or Version 7.x. (All platforms supported in Version 6.x and Version 7.x; Linux must be at Version 6 FixPak 2.)
- DB2 DataJoiner V2.1.1 (AIX, Windows NT, and Solaris Operating Environment).

Procedure:

To migrate an instance:

1. Migrate instances using the **db2imigr** command:

```
DB2DIR/instance/db2imigr [-u fencedID] InstName
```

where

DB2DIR

is /usr/opt/db2_08_01 on AIX and /opt/IBM/db2/V8.1 on all other UNIX-based operating systems.

-u fencedID

Is the user under which the fenced user-defined functions (UDFs) and stored procedures will execute. This parameter is required only when migrating from a client instance to a server.

InstName

Is the login name of the instance owner.

If you have migrated from a non-partitioned version of DB2 to a partitioned version of Enterprise Server Edition, you must update your instances to a partitioned format using the **db2iupdt** command.

The next step in migrating DB2 on UNIX is to migrate existing databases.

Related reference:

- “db2ckmig - Database Pre-migration Tool Command” in the *Command Reference*
- “db2imigr - Migrate Instance Command” in the *Command Reference*
- “db2icrt - Create Instance Command” in the *Command Reference*
- “db2iupdt - Update Instances Command” in the *Command Reference*

Migrating the DB2 Administration Server (DAS)

The task is part of the larger task of Migrating DB2. If you have created a DB2 tools catalog on your DB2 Version 8 system and want to use your existing pre-Version 8 scripts and schedules (for the Control Center) that were created in the pre-Version 8 DB2 Administration Server (DAS), you must migrate the DAS to Version 8.

On Windows, this migration is done automatically if you created the DB2 tools catalog during the installation of Version 8. If you created the DB2 tools catalog after installation, this migration must be done manually.

On UNIX, this migration must be done manually after the DB2 tools catalog has been created, either during the installation or at a later time.

Prerequisites:

You must have:

- An existing DB2 tools catalog.
- DASADM authority to migrate the pre-Version 8 information into the DB2 tools catalog.

Procedure:

To migrate a pre-Version 8 DAS to the DB2 tools catalog, enter the command:

```
dasmigr previous_das_name new_das_name
```

where *previous_das_name* represents the name of the pre-Version 8 DAS instance, and *new_das_name* represents the name of the new Version 8 DAS.

Related tasks:

- “Migrating DB2 (Windows)” in the *Quick Beginnings for DB2 Servers*
- “Migrating DB2 Personal Edition (Windows)” on page 169
- “Migrating DB2 Personal Edition (Linux)” in the *Quick Beginnings for DB2 Personal Edition*

Related reference:

- “dasmigr - Migrate the DB2 Administration Server Command” in the *Command Reference*

Migrating databases

This task is part of the main task of *Migrating DB2*.

Prerequisites:

You require SYSADM authority.

Restrictions:

Migration is only supported from:

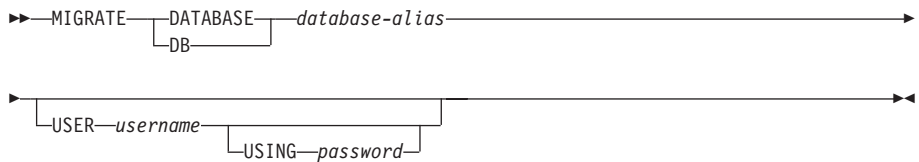
- DB2 Version 6.x or Version 7.x. (all platforms supported in Version 6.x and Version 7.x)
- DB2 DataJoiner V2.1.1 (AIX, Windows NT, and Solaris Operating Environment).

Procedure:

To migrate a DB2 database:

1. Migrate the database using the **db2 migrate database** command.

DB2 MIGRATE DATABASE command



where:

DATABASE *database-alias*

Specifies the alias of the database to be migrated to the currently installed version of the database manager.

USER *username*

Identifies the user name under which the database is to be migrated.

USING *password*

The password used to authenticate the user name. If the password is omitted, but a user name was specified, the user is prompted to enter it.

2. *Optional*: Update statistics. When database migration is complete, old statistics that are used to optimize query performance are retained in the catalogs. However, DB2 Version 8 has statistics that are modified or do not exist in DB2 Version 6 or DB2 Version 7. To take advantage of these statistics, you may want to execute the **runstats** command on tables, particularly those tables that are critical to the performance of your SQL queries.
3. *Optional*: Rebind packages. During database migration, all existing packages are invalidated. After the migration process, each package is

rebuilt when it is used for the first time by the DB2 Version 8 database manager. You can run the **db2rbind** command to rebuild all packages stored in the database.

4. *Optional:* Revoke EXECUTE privileges on external stored procedures that contain SQL data access from PUBLIC. During database migration, EXECUTE privileges are granted to PUBLIC for all existing functions, methods, and external stored procedures. This will cause a security exposure for external stored procedures that contain SQL data access which allow users to access SQL objects for which they would not otherwise have privileges. Revoke the privileges by entering the **db2undgp - r** command.
5. *Optional:* Migrate DB2 Explain tables
6. *Optional:* If you recorded configuration settings before migration, you might want to compare pre-migration configuration settings to current configuration settings to verify successful migration. Verify:
 - database configuration parameter settings
 - database manager configuration parameter settings
 - tablespaces records
 - packages records

Related tasks:

- “Recording system configuration settings before DB2 migration” on page 149
- “Migrating Explain tables” in the *Quick Beginnings for DB2 Servers*

Related reference:

- “MIGRATE DATABASE Command” in the *Command Reference*
- “LIST DATABASE DIRECTORY Command” in the *Command Reference*
- “db2rbind - Rebind all Packages Command” in the *Command Reference*

Migrating DB2 Personal Edition

The sections that follow describe how to migrate DB2 Personal Edition on Windows-based platforms.

Migrating DB2 Personal Edition (Windows)

This topic describes the steps required to migrate from a previous version of DB2 Personal Edition on Windows.

Migrating from a previous version of DB2 requires that you perform pre-installation and post installation tasks.

Prerequisites:

Before you start the migration process, ensure that your system meets the installation requirements for DB2 version 8.

Restrictions:

Migration is only supported from DB2 version 6.x or DB2 version 7.x.

Procedure:

To migrate from a previous version of DB2 Personal Edition (Windows):

1. Prepare to migrate DB2 Personal Edition (Windows).
2. Install DB2 Personal Edition (Windows).
3. Migrate databases on DB2 Personal Edition (Windows).

Related tasks:

- “Preparing to migrate DB2 Personal Edition (Windows)” on page 170
- “Installing DB2 Personal Edition (Windows)” on page 97
- “Migrating databases on DB2 Personal Edition (Windows)” on page 173
- “Migrating DB2 (Windows)” in the *Quick Beginnings for DB2 Servers*

Related reference:

- “Installation requirements for DB2 Personal Edition (Windows)” on page 100

Preparing to migrate DB2 Personal Edition (Windows)

Preparing to migrate DB2 Personal Edition (Windows) is part of the larger task of *Migrating DB2 Personal Edition (Windows)*.

This topic describes the steps required to prepare for migration from a previous version of DB2 Personal Edition on Windows.

Prerequisites:

- To backup a database, you require SYSADM, SYSCTRL, or SYSMANT authority for the database.

Restrictions:

Migration is only supported from DB2 version 6.x or DB2 version 7.x.

Procedure:

To prepare your system for migration:

1. Ensure that all databases you want to migrate are cataloged. To view a list of all the cataloged databases in the current instance, enter the following command:

```
db2 list database directory
```

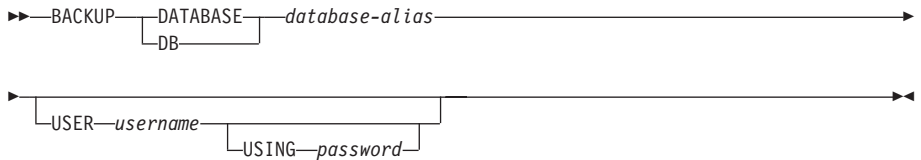
2. Disconnect all applications and users. To get a list of all database connections for the current instance, enter the **db2 list applications** command. If all applications are disconnected, this command will return the following message:

```
SQL1611W No data was returned by the Database System Monitor.  
SQLSTATE=00000
```

You can force a disconnection of applications and users by issuing the **db2 force applications** command.

3. Back up each of your local databases using the **backup database** command:

BACKUP Command



where:

DATABASE *database-alias*

Specifies the alias of the database to back up.

USER *username*

Identifies the user name under which to back up the database.

USING *password*

The password used to authenticate the user name. If the password is omitted, the user is prompted to enter it.

4. Stop the DB2 License Service by entering the **db2licd -end** command.
5. On Windows 2000, the properties of a service can be set so that it restarts if the service fails. If the *restart on failure* option is set for any DB2 services, it must be disabled before proceeding.
6. Stop all command line processor sessions by entering the **db2 terminate** command in each session that was running the command line processor.
7. When all applications and users are disconnected and you have backed up your databases, stop the database manager by entering the **db2stop** command.

8. Enter the **db2ckmig** command to verify that databases owned by the current instance are ready to be migrated. The **db2ckmig** command is located in the `\db2\common` directory on your DB2 version 8 product CD-ROM. The **db2ckmig** command ensures that:

- A database is not in an inconsistent state
- A database is not in backup pending state
- A database is not in rollforward pending state
- Tablespace are in a normal state

DB2CKMIG command

```
▶▶ db2ckmig database_alias /l drive:\path\filename  
      /e  
      /u userid /p password
```

where:

database_alias

Specifies a *database_alias* name of a database to be verified for migration. This parameter is required if the `/e` parameter is not specified.

`/e`

Specifies that all cataloged databases are to be verified for migration. This parameter is required if the *database_alias* parameter is not specified.

`/l drive:\path\filename`

Specifies a drive, target path and filename to keep a list of errors and warnings generated for the scanned database. The path variable is optional; if you do not specify a path, the path from which you execute the `db2ckmig` command will be used. You must specify a filename.

`/u userid`

Specifies the user account used to connect to the database. This parameter must be specified if you are logged on as a user without connect authority.

`/p password`

Specifies the password of the user account used to connect to the database. This parameter must be specified if you are logged on as a user without connect authority.

Your next step is *Installing DB2 Personal Edition (Windows)*.

Related concepts:

- “System administration authority (SYSADM)” in the *Administration Guide: Implementation*

Related tasks:

- “Installing DB2 Personal Edition (Windows)” on page 97

Related reference:

- “BACKUP DATABASE Command” in the *Command Reference*
- “db2ckmig - Database Pre-migration Tool Command” in the *Command Reference*

Migrating databases on DB2 Personal Edition (Windows)

This topic describes the steps required to be taken after installing to complete migration from a previous version of DB2 Personal Edition on Windows. For more in-depth migration instructions and complete command information for **migrate**, refer to the related reference sections at the end of this topic.

Prerequisites:

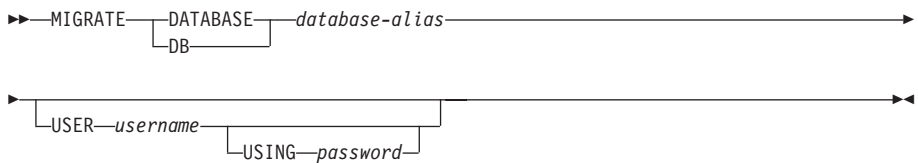
- To migrate a database, you require SYSADM authority.

Procedure:

Once DB2 Personal Edition has been installed, you must complete the migration process by migrating your databases. To migrate databases:

1. Log in with a user account that has SYSADM authority and migrate your databases using the **db2 migrate database** command.

DB2 MIGRATE DATABASE command



where:

DATABASE *database-alias*

Specifies the alias of the database to be migrated to the currently installed version of the database manager.

USER *username*

Identifies the user name under which the database is to be migrated.

USING *password*

The password used to authenticate the user name. If the password is omitted, but a user name was specified, the user is prompted to enter it.

2. *Optional*: Update statistics. When database migration is completed, the old statistics that are used to optimize query performance are retained in the catalogs. However, DB2 Version 8 has statistics that are modified or do not exist in DB2 version 6 or DB2 version 7. To take advantage of these statistics, you may want to execute the **runstats** command on tables, particularly those tables that are critical to the performance of your SQL queries.
3. *Optional*: Rebind packages. During database migration, all existing packages are invalidated. After the migration process, each package is rebuilt when it is used for the first time by the DB2 version 8 database manager. Alternatively, you can run the **db2rbind** command to rebuild all packages stored in the database.
4. *Optional*: Revoke EXECUTE privileges on external stored procedures that contain SQL data access from PUBLIC. During database migration, EXECUTE privileges are granted to PUBLIC for all existing functions, methods, and external stored procedures. This will cause a security exposure for external stored procedures that contain SQL data access which allow users to access SQL objects for which they would not otherwise have privileges. Revoke the privileges by entering the **db2undgp - r** command.

Note: During migration, the database configuration parameter *maxappls* is set to automatic. If you want it set to a different value, you should update it manually.

Related concepts:

- “System administration authority (SYSADM)” in the *Administration Guide: Implementation*

Related reference:

- “MIGRATE DATABASE Command” in the *Command Reference*

Chapter 6. Adding Existing DB2 Servers to a Satellite Environment

Adding Existing DB2 Servers to a Version 8 Satellite Environment	175	Adding New and Existing DB2 Servers to the Satellite Environment Using a Fix Batch	179
Sample Scenario for Adding Existing DB2 Servers to a Satellite Environment	176	Setting the Execution Starting Point to the Next Batch Step	182
Adding Existing DB2 Servers to the Satellite Environment Using Your Own Scripts	178		

The sections that follow describe how to add existing DB2 servers to a satellite environment so they can be administered as members of a group.

Adding Existing DB2 Servers to a Version 8 Satellite Environment

When you install DB2 Universal Database Version 8 on a system that already has a production DB2 environment, the pre-Version 8 code is replaced, and the DB2 environment is migrated to Version 8. Because the system was already in production, it will be at a different state than a newly installed Version 8 satellite that has not yet synchronized. For example:

- One or more databases may have been created on the existing system.
- Tables, indexes, and other database objects to support a business application will have been created and loaded with data.
- Tuning of the system for optimal performance may have been completed; this could involve setting database manager and database configuration values.

By contrast, a newly installed satellite will not have the same level of customization. When new satellites first synchronize, they are customized by their group batches. This customization is not required by existing DB2 systems that are added to the satellite environment.

Restrictions:

If you want to add an existing DB2 system to a satellite environment that is administered by a Version 8 Satellite Administration Center, you must migrate the existing DB2 system to DB2 Version 8.

Procedure:

After you migrate the existing DB2 system to Version 8, you need to do the following when adding this system to an existing group of satellites:

- Migrate databases to the Version 8 format. Please note that the utility, `db2ckmig`, should have been run against any previous databases prior to Version 8 migration.
- Rebind packages. All packages are invalidated during database migration, and under normal operating conditions they will be rebound at first use. Rebinding, however, takes time. To avoid your users waiting for a rebind to complete, you can use the `db2rbind` command to rebind all existing packages.
- Adjust the database manager or database configuration values, as required.

The objective is to make the newly added satellites similar to the other satellites of the group, then have them execute their group batches when they synchronize. You can:

- Use your own scripts to make the migrated DB2 systems similar to the other group satellites
- Use a fix batch to make the migrated DB2 systems similar to the other group satellites

Related concepts:

- “Migration recommendations” on page 146

Related tasks:

- “Adding Existing DB2 Servers to the Satellite Environment Using Your Own Scripts” on page 178
- “Adding New and Existing DB2 Servers to the Satellite Environment Using a Fix Batch” on page 179

Related reference:

- “Migration restrictions” on page 144

Sample Scenario for Adding Existing DB2 Servers to a Satellite Environment

Assume that you have a number of systems that are running DB2[®] Workgroup Edition Version 7.2, and you want to move them to a satellite environment. Also assume that you are going to implement new satellites that run DB2 Universal Database Workgroup Server Edition Version 8 into the same group (as they will run the same application). Compare the state of these systems:

State of the Satellite	Satellite Migrated from Version 7.2	Newly Installed Version 8 Satellite
Database needs migration	Yes	No

State of the Satellite	Satellite Migrated from Version 7.2	Newly Installed Version 8 Satellite
Database definitions require setup	No	Yes
Database and database manager configuration values require customization	Yes - to use new DB2 Version 8 features.	The database and database manager configuration values, which are optimized for Version 8, are set up when the system is installed.
Packages are invalid and need to be rebound	Yes	No
Bind application packages if needed	No	Possibly

Assume that once both types of satellites have reached a common state, they will then run group batches to accomplish ongoing administration.

Every pre-Version 8 DB2 system that is added to a Version 8 satellite environment will need to have its database migrated to the Version 8 format. You may want to perform other optional post migration activities. You can accomplish this in one of two ways and you will have to decide which method is more appropriate for your environment:

- You can use your own scripts. You can develop scripts and run them using your own methods after the existing DB2 system has been migrated to DB2 Version 8.
- You can use a fix batch. A fix batch allows you to determine which systems have performed the migration and to determine its success by examining the script execution. When you have verified that a successful migration has been completed, you can then set the satellite to begin running its group batches.

You should use a fix batch because the results on each system can be easily tracked using the Satellite Administration Center.

Related concepts:

- “Fix Batches” on page 219

Related tasks:

- “Adding Existing DB2 Servers to a Version 8 Satellite Environment” on page 175
- “Adding Existing DB2 Servers to the Satellite Environment Using Your Own Scripts” on page 178

- “Adding New and Existing DB2 Servers to the Satellite Environment Using a Fix Batch” on page 179

Adding Existing DB2 Servers to the Satellite Environment Using Your Own Scripts

Situations may exist where you will use your own scripts, and an established technique to execute them, to migrate pre-Version 8 databases to the Version 8 format, and to perform any other customization that is required to add an existing DB2 system to a Version 8 satellite environment. This newly added satellite will both have to join an existing group and start executing its group batches.

Restrictions:

If you want to add an existing DB2 system to a satellite environment that is administered by a Version 8 Satellite Control Server, you must migrate the existing DB2 system to Version 8.

Procedure:

To add an existing DB2 server to an existing group of satellites:

1. Run your scripts on the DB2 systems that you want to add to a Version 8 satellite environment.
2. Identify the group and application version to which the migrated satellites will belong.
3. Use the Satellite Administration Center to create the DB2 systems that you want to add to the environment as satellites. Create these satellites in the group that they will belong to. You should specify a subgroup for these migrated systems, or have some way of easily identifying them. Using a subgroup, for example, added, you can filter the subgroup for multiselect actions, such as enable, or view these migrated satellites in the satellite details view. When the satellites are created, they are automatically placed in disabled state, and cannot run batches when they synchronize.
4. Set the Execution Starting Point for each of the migrated satellites to the step in the group’s setup, update and cleanup batches at which these systems will start the script execution on first synchronization.
5. Enable the satellites to run batches by multiselecting for the added subgroup and selecting the **Enable** action. Please note that this step is only required if you did not enable the satellites when you created them.
6. Run a test synchronization on the satellite (**db2sync -t**) to ensure that the satellite is set up to synchronize

The newly added satellite will now run group batches each time it synchronizes.

Related concepts:

- “Groups in a Satellite Environment” on page xvi
- “Group Batches” on page 197

Related tasks:

- “Synchronizing the Model Office to Test the Group Batches” on page 273

Adding New and Existing DB2 Servers to the Satellite Environment Using a Fix Batch

You can add pre-Version 8 DB2 servers to a Version 8 satellite environment using a fix batch, and, at the same time, add new Version 8 satellites to the same group. You can use two approaches to complete this task:

- The pre-Version 8 satellites could run fix batches and the new satellites could run group batches to reach the common state.
- The new satellites could run fix batches and the pre-Version 8 satellites could run group batches to reach the common state.

You should consider the first approach. The satellites to be migrated to Version 8 and added to the Version 8 satellite environment are known, and can be tracked until all of them have run the fix batch. As these satellites complete the execution of the fix batch, promote them to execute their group batches. To do this, you would set the execution point at which they are to begin executing their group batches. You may want to place the migrated and newly added satellites in a special subgroup, for example added, when they are created so that their progress can easily be tracked using the sort and filter capabilities of the Satellite Administration Center.

Using this technique to handle satellites that are migrated to DB2 Version 8, new satellites can be added to the group at any time, as the group batches they would run would always perform the initial customization for them, without intervention from an administrator to set them up to run a special fix batch.

You would create an application version for the group, and associate a setup batch with the application version. The setup group batch would contain the following steps to accomplish the initial customization of a new system. Any new Version 8 satellite would execute this batch the first time it synchronized. The setup batch would:

1. Create the database on the satellite.

2. Customize the database and database manager configuration values as required.
3. Define any required indexes.

Restrictions:

If you want to add an existing DB2 system to a satellite environment that is administered by a Version 8 Satellite Control Server, you must migrate the existing DB2 system to Version 8.

Procedure:

To add the systems migrated from a pre-Version 8 environment to a Version 8 satellite environment using a fix batch, have the fix batch perform the following steps:

1. Run the **db2 migrate database xxx** command to migrate each database (where 'xxx' represents the database to be migrated)
2. Update the database and database manager configuration values as required, to take advantage of new Version 8 function.
3. Rebind the packages

The targets of the script execution must be cataloged on the satellite. The instance and database names must match those on the Control Center's instance, as these will have been used to name execution targets when the targets are created using the Satellite Administration Center. In addition to other migration activities, the fix batch may have to catalog additional node and database directory entries so that all the targets of script execution are defined on the migrated satellite.

The systems migrated to DB2 Version 8 have to be created as satellites in the Satellite Administration Center, and have to be set up to run the fix batch on their initial synchronization. When this fix batch is run, these satellites can be promoted to run group batches and the initial step that they would run in each batch can be set.

To add the pre-Version 8 DB2 systems to a Version 8 satellite environment and have them begin to execute their group batches:

1. Identify the group and application version to which the pre-existing DB2 systems will belong.
2. Use the Satellite Administration Center to create the DB2 systems that you want to add to the environment as satellites. Create these satellites in the group that they will belong to. You should specify a subgroup for these existing DB2 systems, for example, added, or have some way of easily identifying them. Using a subgroup, you can filter the subgroup for

multi-select actions, such as enable, or view these added satellites in the satellite details view. When the satellites are created, they are automatically placed in disabled state, and cannot run batches when they synchronize.

3. Create a fix batch that contains the required steps. This fix batch should be thoroughly tested before it is made available for execution by the production systems that are being added to the satellite environment.
4. Put each satellite in the subgroup in fix mode and specify the fix batch that will be run.
5. Enable the satellites to run batches by multiselecting for the added subgroup and selecting the **Enable** action.
6. Migrate each system to DB2 Version 8.
 - a. Install DB2 Universal Database Version 8 on each of the existing DB2 systems.
 - b. Catalog the SATCTLDB database on the systems. The satellite control database must be cataloged on the satellites before they can synchronize. You can add the catalog entries in two ways:
 - 1) You can run the DB2 Synchronizer application in test mode (**db2sync -t**). You will be prompted to enter the information to access the SATCTLDB database.
 - 2) You can import a client profile file the installation process to catalog the SATCTLDB database using the **DB2.CLIENT_IMPORT_PROFILE** keyword. The client profile should only contain the information to catalog the SATCTLDB database. You can produce this profile from the catalog information in the Control Center's instance using the Client Configuration Assistant.
7. Have each migrated system synchronize:
 - a. Run a test synchronization on the satellite (**db2sync -t**) to ensure that the satellite is set up to synchronize.

Note: Running the DB2 Synchronizer application in test mode will not cause the satellite to execute the fix batch.

- b. Run the DB2 Synchronizer application (in production mode), by running the **db2sync** command. In this situation, the satellite will download and execute the fix batch.

After the fix batch has executed successfully, or produced satisfactory results, the newly added satellite is ready to run its group batches. To have it start executing group batches, promote the satellite to production and set its execution starting point as required.

Note: If the fix batch produced satisfactory results but still failed in execution, you will need to enable the satellite to execute its group batches and promote it.

The newly added satellite will now execute its group batches the next time it synchronizes.

Related concepts:

- “Fix Batches” on page 219
- “Promotion of Test Batch Steps to Production Batch Steps” on page 201

Related tasks:

- “Cataloging Instances and Databases on Production Satellites” on page 237
- “Setting the Execution Starting Point for a Satellite” on page 267
- “Adding Existing DB2 Servers to a Version 8 Satellite Environment” on page 175
- “Adding Existing DB2 Servers to the Satellite Environment Using Your Own Scripts” on page 178

Setting the Execution Starting Point to the Next Batch Step

When you set the execution starting point for a satellite, you can specify that the satellite begin executing a group batch (setup, update, cleanup) after the last batch step in the batch. This action prevents the satellite from executing any existing batch steps for that batch. You may want to do this if, for example, you add an existing DB2 system into the satellite environment. In this situation, the satellite already has a database, database objects, and data, and you will likely use a fix batch to modify the newly added DB2 system to the configuration that you want (if necessary). You will not want this new satellite to execute the setup batch to create the database definition.

Restrictions:

You cannot specify that a satellite execute a fix batch from the next batch step.

Procedure:

To prevent the satellite from executing the batch steps of a group batch, use the Set Execution Starting Point window. When you specify the execution starting point for the satellite, select the next batch step from the group batch. In this way, the satellite will not execute any of the existing batch steps of the group batch when it synchronizes.

If you later add one or more new batch steps to the group batch, the satellite will execute these batch steps. That is, the next batch step is always after the last batch step in a group batch.

Related concepts:

- “Group Batches” on page 197
- “Fix Batches” on page 219

Related tasks:

- “Setting the Execution Starting Point for a Satellite” on page 267
- “Setting the execution starting point for a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Part 2. Administering a Satellite Environment

Chapter 7. Batches and Application Versions

Batches	187	Test, Production, and Obsolete States of	
Batch Steps	188	Application Version Levels	208
Components of a Batch Step	188	States of an Application Version	213
Parameterized Scripts	193	Update Batch in a Test Level of an	
Batch Modes	194	Application Version	213
Application Versions in a Satellite		Promoting a Test Level to a Production Level	
Environment	196	in an Application Version	214
Group Batches	197	Creating a Test Level from a Production	
Group Batches in the Test-Production Cycle	199	Level in an Application Version	215
Execution of Test Batch Steps by Test		Obsoleting a Production Level in an	
Satellites	200	Application Version	216
Promotion of Test Batch Steps to Production		Relationships Between Batches and Batch	
Batch Steps	201	Steps	217
Life Cycle of an Application Version	201	Storage of Script on a Satellite During a	
Example Test-Production Cycle of a Setup		Synchronization Session.	218
Batch	204	Fix Batches	219
Levels of an Application Version.	206		

The sections that follow describe batches and application versions in a satellite environment, and how you use them to administer satellites.

Batches

You use batches to ensure that the satellites in a group remain as similar as possible. A batch is an ordered set of one or more batch steps.

All group satellites that are at the same application version execute the same group batches. These batches set up and maintain the database definition for the version of the application on these satellites.

Group satellites also execute the batch steps of these batches in the same order. Because the group satellites execute the same group batches and batch steps, each satellite in the group (and with the same application version) will be similar.

You can also use batches to fix satellites that either report problems or require an adjustment. The function of a batch depends on its mode.

Related concepts:

- “Batch Steps” on page 188
- “Components of a Batch Step” on page 188

- “Fix Batches” on page 219

Batch Steps

You create batch steps to set up and maintain the database definition for the application version. Batch steps are executed on the satellite when the satellite synchronizes.

A batch step is made of the following components:

- A *script*. The script can be one or more DB2® commands, SQL statements, or operating system commands that you want a satellite to run.
- An *execution target*. The scripts that you create can execute against a DB2 instance, DB2 database, or on the operating system on the satellite. The DB2 instance, DB2 database, or operating system against which the script executes is known as an execution target.
- An *authentication credential*. Before a script can execute against a DB2 instance or DB2 database, it must be authenticated. That is, the script requires the combination of a user ID and password so that the satellite can attach to the instance or connect to the database. This combination of user ID and password is known as authentication credentials.
- A *success code set*. The execution of a script is considered to be successful if its return code is within a set of return codes that you predefine for that script. This set of codes is known as a success code set.

The satellites always execute the batch steps within a batch in the sequence in which they appear in the batch. When one batch step within a batch is executed successfully, the next batch step is executed. If, as defined by the success code set, an error occurs when a satellite is executing a batch step, that satellite stops executing its group batches, and reports an error back to the satellite control server. When the error is fixed, the satellite can continue executing from the batch step that caused the error.

Related concepts:

- “Application Versions in a Satellite Environment” on page 196
- “Components of a Batch Step” on page 188
- “Fix Batches” on page 219

Components of a Batch Step

A batch step is the combination of a script, an execution target, authentication credentials, and a success code set.

Script The script component of a batch step can be one or more DB2®

commands, SQL statements, or operating system commands. The script can be a sequence of one or more commands or statements.

You can create scripts manually or use the wizards, notebooks, and windows in the Control Center to create scripts and save them to the Task Center. If the scripts that you want are in the Task Center, you can import them into the group batches of an application version by using the Change Level notebook.

Some scripts that are used in group batches must be parameterized either to take into account differences between satellites that belong to the same group or to identify values that might be independent of the satellite that is synchronizing. Parameterizing a script means adding an embedded marker into a script that will be replaced with an attribute when the satellite synchronizes.

For scripts that execute against DB2 instances or DB2 databases, all the commands or statements in the script must execute against a single target. For example, if the script has a target of a DB2 instance called `test`, all of the commands in the script will be executed against `test`. Similarly, if the target is a DB2 database called `payroll`, all of the commands and SQL statements in the script will be executed against `payroll`.

When you specify a DB2 command and its target is either a DB2 instance or a DB2 database, you do not have to specify the DB2 command prefix, nor do you have to explicitly attach to the instance or connect to the database within the script. The instance attachment or database connection is automatically performed based on the target that is specified for the script. For example, to list the tables of a target DB2 database, the script would contain the following DB2 command:

```
LIST TABLES FOR ALL;
```

Operating system scripts can contain commands that execute against the operating system, as well as DB2 commands that execute against instances or databases.

Note: An operating system script must be either a batch script or a command script. That is, the script must have an extension of either `.bat` or `.cmd`. If you want to execute a script, such as a Perl script, that does not have an extension of `.bat` or `.cmd`, that script must be inside the batch or command script.

If you want to include in an operating system script DB2 commands that execute against instances or databases, ensure that the script includes the command to explicitly attach to an instance or connect to a database before the DB2 command or SQL statement is executed.

This will ensure that the DB2 command or SQL statement in the script is executed against the intended target instead of the default instance or database. Also, you must use the DB2 prefix with the command. For example, to list the tables of a DB2 database, the operating system script would contain the following DB2 commands. Note that you do not use a statement termination character for DB2 commands and statements in an operating system script.

```
DB2 CONNECT TO DATABASE test
DB2 LIST TABLES
DB2 CONNECT RESET
```

Execution target

A script executes locally on a satellite, but, depending on the type of the script, the target can be local or remote. For example, a DB2 command or SQL statement will execute against a target DB2 instance or DB2 database, either of which can be local or remote. An operating system command, however, can only be executed by the command processor of the local operating system. Operating system scripts can be executed against anything that is managed by the operating system (such as programs and file systems).

The execution target defines how the script is launched, so the script cannot be associated with more than one target. For example, an instance attach (using authentication credentials) will be initiated before a script with an execution target type of instance is executed.

Note: If the target of a script is a DB2 instance, an ATTACH statement will be automatically issued to the instance before the script is executed. A DETACH statement is automatically issued when execution is complete. Similarly, if the target is a DB2 database, a CONNECT statement will be automatically issued to the database before the script is executed. When the execution is complete, a CONNECT RESET statement is automatically issued. This means that all DB2 instances and DB2 databases that are targets of scripts must be cataloged at the satellite, regardless of whether they are local or remote to the satellite.

Authentication credential

Authentication credentials are required for DB2 commands and SQL statements to execute against a DB2 instance or DB2 database. In addition, at the start of the synchronization process, the satellite must be able to authenticate against the satellite control server to obtain the scripts that it is to execute. An authentication credential is the combination of user ID and password that is required to attach to an instance or to connect to a database. Each script that has an instance or database execution target is associated with a specific authentication credential. Authentication credentials are not required by operating system scripts.

Success code set

When a script is executed, its success or failure is defined by the associated success code set. For DB2 command or SQL statement scripts, each command or statement is executed individually, and its SQLCODE compared with the success code set associated with the batch step. If the statement is successful, the next statement is executed, and so on. Otherwise, execution of the batch is terminated, subsequent steps in the batch are not executed, and an error is reported to the satellite control server. After the error is reported, synchronization stops, and no additional batches or batch steps are executed. The satellite is disabled from synchronizing and marked as FAILED at the satellite control server. The satellite cannot synchronize again until you fix it.

In the case of operating system scripts, the entire script is executed to completion, then the exit code or return code is compared with the associated success code set. If successful, the next batch step is executed. Otherwise, execution of the batch is terminated, subsequent steps in the batch are not executed, and an error is reported to the satellite control server. After the error is reported, synchronization stops, and no additional batches or batch steps are executed. The satellite is disabled from synchronizing and marked as FAILED at the satellite control server. The satellite cannot synchronize again until you fix it.

A success code set is one or more comparative operators and numeric values. The comparative operators can be =, >, or <. Numeric values can be any positive or negative integer, or zero (0). All members of the success code set are compared to the SQLCODE that is returned by a DB2 command or an SQL statement, or compared to the exit code or return code that is returned by an operating system command. If the SQLCODE, exit code, or return code is within the defined set, execution of the script is considered to be successful.

The following rules apply for success code sets:

- The set can only have one greater than (>) condition, where the associated code must be greater than or equal to (>=) any less than (<) condition that is specified.

For example, if you specify (>, 5) and (<, 0), the error codes for that set are 0, 1, 2, 3, 4, 5. You cannot specify (>5) and (<, 6) because this will provide all numbers.

- The set can only have one less than (<) condition, where the associated code must be less than or equal to (<=) any greater than (>) condition that is specified.

For example, if you specify (<, 0) and (>, 5), the error codes for that set are 0, 1, 2, 3, 4, 5. You cannot specify (<, 5) and (>,4) because this will provide all numbers.

- There can be zero or more unique equals (=) conditions, but no duplicate equals conditions.

The following example shows how to set up a success code set for a script that contains multiple SQL statements. Assume that each of these statements drops a table. Each DROP TABLE statement in the script will return an SQLCODE. DROP TABLE statements can return non-zero SQLCODEs that do not represent an error state. You must determine the set of SQLCODEs that do not indicate an error and include them in the success code set for the script. For example, the following return codes indicate the successful execution of the script (that is, if any of the following conditions are met, execution of the script continues):

```
SQLCODE = 0, SQLCODE > 0, SQLCODE = -204
```

In this success code set, the SQLCODE = 0 indicates that the DROP TABLE statement completed successfully. The SQLCODE > 0 indicates that processing can continue even if a positive SQLCODE is returned. The SQLCODE = -204 indicates that if the table does not exist when the DROP TABLE statement is issued, processing can continue.

You can create the execution target, authentication credentials, and success code set for the batch step by using the Satellite Administration Center. You can create these components of the batch step when you create the batch step, or you can create them in advance, then include them when you create the batch step.

Related concepts:

- “Parameterized Scripts” on page 193
- “Authentication Credentials” on page 221

Related tasks:

- “Changing the level of an application version : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Creating an authentication : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Creating a success code set : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Creating a target : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Parameterized Scripts

When you administer satellites at the group level, certain types of scripts that are used in group batches need to be parameterized so that they can be executed by the satellites. You can parameterize a script to identify a value that is independent of the satellite. This is known as a *table parameter*. You can also parameterize a script to identify a value that is unique to a satellite (such as a user ID). This is known as a *contextual parameter*.

You can add a contextual parameter to a WHERE clause to customize it for the satellites. You can find information that differs from satellite to satellite in the SATELLITES table in the satellite control database. You can also obtain this information from the satellite details view in the Satellite Administration Center.

When the satellite synchronizes, requesting the group batches from the satellite control server, the satellite control server checks whether any script is parameterized before allowing the satellite to download it. If any script is parameterized, the satellite control server will substitute the appropriate table or contextual parameter for the parameterized markers before the satellite downloads the script.

You specify whether a script is parameterized by using the Change Batch Steps notebook and the Create and Edit Script windows, all of which are available from the Satellite Administration Center. There are two types of parameter markers that you can use:

Table Parameters The table parameter is the general mechanism for specifying scalar values. You can use a table parameter to specify a single column value for a single row of a table in the satellite control database.

Syntax

```
{{tablename:colname:predicates}}
```

The syntax translates to `SELECT colname FROM tablename WHERE predicates`. The parameters are as follows:

tablename

Is the fully qualified two-part name of a table in the satellite control database. For example, `schema.tablename`.

colname

Is the name of the column in *tablename* that contains the value to be substituted for the parameter marker.

Note: *colname* cannot resolve to a column with a data type of CLOB, BLOB, GRAPHIC(1), GRAPHIC(n), VARCHAR(n), or LONG VARCHAR.

predicates

Is the WHERE clause predicates that are used to identify the row that contains the required column value. The predicates should identify a single value. If not, one value is returned, but the value is not deterministic.

Contextual Parameters Contextual parameters refer to the values that apply specifically to the satellite that will execute a script. These values are all of the attributes of the satellite recorded in the SATELLITES table for that satellite. The contextual parameter is more restrictive than a table parameter. The *predicates* parameter is implicitly defined to select the specific row that applies to the satellite.

Syntax

{{SATELLITES:colname}}

Where:

colname

Is one of the columns in the SATELLITES table

Note: *colname* cannot resolve to a column with a data type of CLOB, BLOB, GRAPHIC(1), GRAPHIC(n), VARGRAPHIC(n), or LONG VARGRAPHIC.

Examples The examples that follow show how to use parameters:

- To obtain the name of the group to which a satellite belongs:

{{SATELLITES:GROUP}}

- To obtain the last name of the user of a satellite:

{{SATELLITES:LAST_NAME}}

Related tasks:

- “Viewing satellite details : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Editing a script : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Changing Batch Steps : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Batch Modes

There are three different modes of batches in the satellite environment:

Group

To set up and maintain the database definition on satellites, you use group batches. A group batch is associated with a particular

application version. In addition, each satellite is associated with an application version. When a satellite synchronizes, it downloads and executes the group batches for its particular application version. Before the group satellites execute the batch steps, they are all similar. When all the satellites have executed all the batch steps of all their group batches, they remain similar.

Because the satellites execute the group batches to maintain their database definition, you only have to work with the batches, instead of individually maintaining the hundreds, if not thousands of group satellites.

A group batch will be either a setup, update, or cleanup batch. You work with group batches by using the Edit Application Version window in the Satellite Administration Center. You can also create an unassigned batch using the Create Batch window in the Satellite Administration Center, and then assign it to a group when you edit the application version.

Fix A fix batch is one that is created to fix a problem on one or more satellites, so fix batches are not assigned to a specific group or to an application version

Unassigned

An unassigned batch is one that maintains its mode until it is assigned to either of the following:

- An application version, to be executed by a group as a setup, update, or cleanup batch
- A satellite, that will execute the batch as a fix batch

You can modify an unassigned batch in any fashion, and you can also delete it. If you assign an unassigned batch to an application version, the batch becomes a group batch. If you assign the batch as a fix batch, which you use to make changes to a specific satellite, the unassigned batch becomes a fix batch. In both cases, assigning the batch permanently changes the mode of the batch. That is, the batch cannot be changed back to an unassigned batch.

You work with unassigned batches by using the Create Batch and Edit Batch windows in the Satellite Administration Center.

Related concepts:

- “Group Batches” on page 197
- “Application Versions in a Satellite Environment” on page 196
- “Fix Batches” on page 219

Related tasks:

- “Editing an application version : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Creating a batch : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Editing a batch : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Application Versions in a Satellite Environment

Although the satellites of a group run the same application, they do not necessarily run the same version of this application. Each version of the application may require a database definition that is different from that required by other versions of the same application. Each particular version of the application is identified by an *application version*. The application version exists both at the satellite control server, and on the satellite.

You create an application version on the satellite control server using the Create Application Version window of the Satellite Administration Center. When you create an application version on the satellite control server, you supply a unique identifier for the application version.

When you set the application version on the satellite, you specify the value that corresponds to the version of the application that runs on that satellite. You can set the application version on the satellite by using either the `db2SetSyncSession` API or the `db2sync -s` command when the application is being installed and configured; you can also set the application version when installing DB2® on the satellite. (If you are not sure about the value of the application version on the satellite, you can use the `db2sync -g` command to retrieve the value.)

After the application version is created on the satellite control server, you use the Edit Application Version window to associate a setup, update and cleanup group batch with it. These group batches set up and maintain the required database definition to support a particular version of the application. Each application version that exists for a group is associated with its own setup, update, and cleanup group batches. Because the satellites of a group will be running at least one version of the application, the group will have at least one application version.

When a satellite synchronizes, it uploads its application version to the satellite control server. The satellite control server uses this information, in conjunction with the group that the satellite belongs to, to determine which of the group batches the satellite will execute. The satellite control server only allows the satellite to download and execute the group batches that correspond to the satellite’s application version.

At some point in time, you will need to deploy a new version of the application. Typically, a new version of an application will require a different database definition than the previous version. Consequently, the batches and the associated scripts used to maintain the new database definition will be different.

If you have a large number of satellites in a group, you will probably want to stage the deployment of a new version of the application within that group. That is, you will want to keep most satellites of the group at one version of the application, and use a subset of the group satellites to determine whether the new version of the application meets your business requirements. To stage the deployment, however, you may have to support more than one set of batches for a group of satellites. You will have one set of batches for each version of the application that is used by the group. That is, you will have one set for the original version, and one set for the new version to be deployed. The satellite administration environment supports this requirement through the implementation of application versions.

Related tasks:

- “Setting the Application Version on a Satellite” on page 249
- “Installing a New Version of the Application” on page 320
- “Creating an application version : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Editing an application version : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Related reference:

- “db2sync - Start DB2 Synchronizer Command” in the *Command Reference*
- “db2SetSyncSession - Set Satellite Sync Session” in the *Administrative API Reference*

Group Batches

Group batches are associated with an application version, and you use these batches to set up and maintain the database definition on the satellites that are running a particular version of an application. Using group batches allows you to maintain consistency among the satellites of a group without having to maintain each satellite separately.

Even though the satellites in the group may execute the batch steps of the group batches at different times, each satellite of the group will execute the same set of batch steps in the same order. This ensures the consistency of the satellites that belong to the group. It also simplifies the management of large numbers of satellites. You know that the satellites are similar because they

have all executed the same set of batch steps in the same order. If the satellites started out similar before executing the group batches of an application version, they will remain similar after they have all executed the group batches.

Three types of group batches can be executed by satellites when they synchronize: setup, update, and cleanup:

Setup A setup batch is an ordered set of batch steps that is executed first, before any other batches. You typically use the setup batch to set up the database definition for the satellite, the update batch to maintain the data on the satellite, and the cleanup batch to perform cleanup activities on the satellite. Each batch step in a setup batch is run only once by a satellite. If you add a new batch step to the setup batch, and the setup batch has already been executed by a given satellite, only the new batch step will be executed by that satellite. You can use the setup batch to set up the database definition on the satellite, including schemas, tables, indexes, and any other database objects that you require. You can also use the setup batch to set configuration parameter values.

Update

An update batch is an ordered set of batch steps, and each step is executed every time that the satellite synchronizes. This type of batch is run after a setup batch is run and before a cleanup batch is run. The batch steps in an update batch are those that can be executed repeatedly. A typical update batch will consist of a data synchronization batch step.

The steps in an update batch are considered to be idempotent, in the sense that they can be repeatedly executed without changing the current state or database definition of the satellite to a different state with each invocation of the step. For example, a table can be replicated multiple times without resulting in a change in the replication configuration. In contrast, the batch step in the setup batch that originally created the table would be executed only once.

Cleanup

A cleanup batch is an ordered set of batch steps that is executed last, after the update batch. Each batch step in a cleanup batch is run only once by a satellite. If you add a new batch step to the cleanup batch, and the cleanup batch has already been executed by a given satellite, only the new batch step will be executed by that satellite.

A typical cleanup batch is one that contains a batch step that updates the database statistics.

When the satellite synchronizes for the first time, it executes, in order, the batch steps of the setup batch to configure itself, the batch steps of the update batch to populate its tables, and the batch steps of the cleanup batch to perform any cleanup activities. After a satellite synchronizes for the first time, it will execute any new batch steps that are appended to the setup batch to modify its database definition (if required). The satellite then executes all the batch steps of the update batch to maintain its data. Finally, the satellite will execute any batch steps that are appended to the cleanup batch.

Only one of each type of group batch can be associated with an application version. However, depending on your requirements, you might not need to create all three types of group batches. For example, you might decide to use a different mechanism to set up the database definition than the setup batch. If a specific type of batch does not exist when a satellite synchronizes, that batch type is bypassed.

You can create group batches from the Edit Application Version window. You can also create unassigned batches from the Create Batch window, then assign them as group batches when you edit the application version.

Related tasks:

- “Editing an application version : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Creating a batch : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Editing a batch : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Group Batches in the Test-Production Cycle

A group batch can be in one of three states: obsolete, production, and test. A production level consists of group batches, the steps of which are in the production state and are only executed by the production satellites of the group. Given sufficient time, all the satellites of the group will contact the satellite control server to synchronize, download the batch steps that they have to execute, and execute the associated scripts. Consequently, all the group satellites begin at a consistent state, and end at a consistent state. If you find that you require modifications to the database definition, you can create a new level of the application version. The new level will be in test state. You can then modify the setup, update, and cleanup batches, as required, to make the modifications.

When an update batch is copied to a new level, all the batch steps in the new level of the update batch are set to the test state. The new level of the update

batch contains no production batch steps. As a result, you can modify the batch steps of an update batch in any way that you require. You can change batch steps, reorder them, add new ones, and delete existing ones. When a test satellite synchronizes, it runs all the batch steps that are in the test state. By definition, the batch steps of the update batch are idempotent, and satellites always execute all the batch steps of the update batch when they synchronize.

Related concepts:

- “Update Batch in a Test Level of an Application Version” on page 213

Execution of Test Batch Steps by Test Satellites

A test satellite executes the batch steps of a setup or cleanup batch differently than those of an update batch. When you create a test level of an application version from a production level, all the batch steps in the setup and cleanup batches are copied, but they remain at the production state. You cannot modify these production batch steps in any way, nor can you reorder or delete them. You can only append new test batch steps to them. Because the appended batch steps are in the test state, you can modify, reorder, or delete them.

The next time the test satellites synchronize, the satellite control server will recognize that the test satellites have not executed the new test batch steps. The test satellites will only download and execute those test batch steps that they have not previously executed. A test satellite will not download and execute any of the production batch steps, unless that test satellite has not already executed the full set of production batch steps. This situation can occur with a new test satellite.

Typically, you will want to verify the changes that the test batch steps implement on a small number of test satellites before promoting the associated test level to production. If changes are required, you can modify, reorder, and add or delete the test batch steps and test them again. Because you can test repeatedly, you can refine the test batch steps until you are satisfied with the results that they generate on the test satellites. This process occurs without any impact to the production satellites.

Promotion of Test Batch Steps to Production Batch Steps

When you are satisfied with the results generated by the test batch steps, you promote the test level to production. The next time a production satellite synchronizes, it will download and execute the new batch steps in the setup and cleanup batches, and all of the batch steps in the update batch, whether they are changed or not.

The interrelationship between the state of a batch step (test or production), and the sequence of its execution within a group batch, is pivotal to the test-production cycle of batch step development.

Life Cycle of an Application Version

The description that follows describes the test/production/obsolete development model, and how this model applies to the levels of an application version through the life cycle of the application version.

When you create the first level of a new application version, that new level is identified as level 0 and is created in the test state. In the example that follows, there are two batches: `configure`, which sets up the database definition, and `datasync`, which maintains the data used by the application:

Application Version				
Level	State	Setup Batch	Update Batch	Cleanup Batch
0	Test	<code>configure</code>	<code>datasync</code>	

You use your test satellites to fully test the batches associated with this level of the application version. When you are satisfied with the results on your test satellites, you promote this level to production. (If you are not satisfied with the results, you can continue modifying the batches until they produce the results that you want. You can also delete level 0, create level 0 again, and create new batches for level 0). When level 0 is promoted, your production satellites can execute the batches of this level. Level 0, which was at the test state, then moves to the production state, as follows:

Application Version				
Level	State	Setup Batch	Update Batch	Cleanup Batch
0	Production	<code>configure</code>	<code>datasync</code>	

Assume that you are in full production, and identify a performance problem. To address this problem, you decide to add an index to one of the tables. You

want to test this database definition change on your test satellites before making it available to your production satellites. To do this, start by creating a test level of your existing production level. This new level is in the test state and is a copy of your existing production level. Then, edit level 1 and append a new batch step to the configure batch to create the index:

Application Version				
Level	State	Setup Batch	Update Batch	Cleanup Batch
0	Production	configure	datasync	
1	Test	configure (with batch step added to create index)	datasync	

Again, you use your test satellites to fully test the changes associated with the new test level, level 1. Because all members of your test satellites already executed all the batch steps associated with level 0 when it was in the test state, the test satellites will only execute the new batch step in the configure batch to create the index when they next synchronize. (They will also execute the entire datasync batch.) When you are satisfied with the results of the index creation, promote level 1 to production. Then your production satellites can execute the batches of this level. Level 0 is no longer adequate to support the end-user application. Because only one database definition can be in use in the production environment, level 0 is obsolete:

Application Version				
Level	State	Setup Batch	Update Batch	Cleanup Batch
0	Obsolete	configure	datasync	
1	Production	configure (with batch step added to create index)	datasync	

When they next synchronize, the production satellites that have previously synchronized execute only the create index batch step of the configure batch. Satellites that are synchronizing for the first time execute all the batch steps of the configure batch, including the create index batch step. To maintain the end-user application data, all satellites run the datasync batch after they execute the configure batch.

Again, time passes. Although your tables are indexed, the performance of the application is slowly becoming worse. You decide that you need to perform data reorganization on the tables on the satellite to reduce data fragmentation. Again, you need to create a test level from your production level. You decide

to add a cleanup batch to contain the batch step that reorganizes the data. In this way, the data is reorganized after it is updated by the datasync batch:

Application Version				
Level	State	Setup Batch	Update Batch	Cleanup Batch
0	Obsolete	configure	datasync	
1	Production	configure (with batch step added to create index)	datasync	
2	Test	configure (with batch step added to create index)	datasync	reorganize

Again, you test level 2 with your test satellites. When you are satisfied with results, you promote level 2 to production. Level 1 is automatically obsoleted:

Application Version				
Level	State	Setup Batch	Update Batch	Cleanup Batch
0	Obsolete	configure	datasync	
1	Obsolete	configure (with batch step added to create index)	datasync	
2	Production	configure (with batch step added to create index)	datasync	reorganize

Assuming no new satellites join the group, all satellites bypass executing the configure batch, because they have already executed all of its steps. The satellites start by executing the datasync batch to maintain the data for the end-user application. When the satellites complete executing the datasync batch, they run the reorganize batch to reorganize the table data. Each satellite only executes the batch step in the reorganize batch once.

Related tasks:

- “Promoting a Test Level to a Production Level in an Application Version” on page 214
- “Creating a Test Level from a Production Level in an Application Version” on page 215
- “Creating an application version : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Example Test-Production Cycle of a Setup Batch

The following example shows a single cycle of the test-production cycle. This example illustrates the development of a setup batch.

1. Assume that you have the following batch steps in production in the setup batch for level 0 of an application version. These batch steps set up the database definition.

Setup Batch of Production Level 0				
Batch Step	Script	Batch Step State	Success Code Set	Execution Target
1	Create database	Production	Create database success code set	Local satellite instance
2	Create table A	Production	Create table success code set	Local satellite database
3	Create table B	Production	Create table success code set	Local satellite database

Over time, the tables on the satellites have more and more rows added to them, and the application does not perform as well as it did when it was first implemented. You realize that if one of the tables had an index, the application would perform better. But, if you add an index, you should also adjust the heap size on the satellites to account for the new index.

2. The first step is to use the Edit Application Version window to create a test level of the application version from the production level. Then update the test level of the setup batch and append the batch steps that both create the index and change the heap size:

Setup Batch of Test Level 1				
Batch Step	Script	Batch Step State	Success Code Set	Execution Target
1	Create database	Production	Create database success code set	Local satellite instance
2	Create table A	Production	Create table success code set	Local satellite database
3	Create table B	Production	Create table success code set	Local satellite database
4	Create index on table A	Test	Create index success code set	Local satellite database
5	Heap size alterations	Test	Change configuration parameter success code set	Local satellite database manager

Two steps (4, 5) are added in sequence, immediately following the three pre-existing production batch steps. Because the new batch level is in the test state, only test satellites can download and execute the new batch steps in it. Production satellites will continue to execute the production-level setup batch, if required. During the testing of these new batch steps, you can modify the test batch steps as necessary to prepare them for production.

Because the test satellites already executed steps 1 through 3 of the setup batch when it was in the test state, they will begin executing the new test-level setup batch at step 4 (satellites only execute the steps of a setup batch once, and in the sequence that they appear in the batch). If you find that the results of the new batch steps are not satisfactory:

- Correct the test batch step that produced the incorrect results. You can have the test satellite re-execute this batch step and any that follow by changing the step at which the test satellite starts executing the test batch. To specify the step from which a satellite begins executing a batch, use the Batches page of the Edit Satellite notebook, which is available from the Satellite Administration Center.
 - In some situations, you must undo the results of the test batch step or steps. Assume that neither step 4 nor step 5 of the setup batch produce the required results. In this situation, you can use a fix batch to undo the results of steps 4 and 5. You then correct the test batch steps, and set the execution start point for the test satellite so that it re-executes the test batch steps.
 - You can also use a fix batch to undo not only the effects of the test batch steps, but also of all the batch steps that the test satellite has previously executed. In this situation, after you fix the test batch steps, you set the execution starting point to step 1, and the test satellite executes all the production and test batch steps in sequence.
3. When you are satisfied with the results of the test, you promote level 1 to production. The setup batch steps change to the production state. You do this by using the Edit Application Version window, which is available from the Satellite Administration Center. The next time that the production satellites synchronize, they will download and execute the new batch steps, 4 and 5.

Setup Batch of Test Level 1				
Batch Step	Script	Batch Step State	Success Code Set	Execution Target
1	Create database	Production	Create database success code set	Local satellite instance
2	Create table A	Production	Create table success code set	Local satellite database

Setup Batch of Test Level 1				
3	Create table B	Production	Create table success code set	Local satellite database
4	Create index on table A	Production	Create index success code set	Local satellite database
5	Heap size alterations	Production	Change configuration parameter success code set	Local satellite database manager

Related concepts:

- “Fix Batches” on page 219

Related tasks:

- “Identifying the Failed Satellite” on page 337
- “Editing an application version : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Editing a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Levels of an Application Version

Assume that you have a large group of satellite running the first version of the application. Because the group is large, it may not be practical to perform the deployment at one time. Because of this, you stage the deployment.

To stage the deployment, you enable satellites that have a shared characteristic, such as a common subgroup, to begin executing the group batches. As you deploy more and more satellites, scaling the environment larger and larger, you may find that the database definition that seemed appropriate for your application in the earlier stages of the deployment is no longer adequate. For example, the performance of the application is becoming worse. It may be that the amount of data that is maintained by each satellite is larger than you originally anticipated. This situation may not necessarily result in satellites returning errors and requiring fix batches, but you need to make some changes to the database definition. In this situation, adding indexes to one or more tables will improve the performance of the application. Rather than creating a new application version, which is appropriate only if you are going to change the version of the application, you would create a new level of the *existing* application version. You then modify this new level of the application version to change the database definition.

A new level of an application version is a copy of the setup, update and cleanup batches of the previous level of the application version. You can add one or more additional batch steps that you design to modify the database definition. Each level of an application version is associated with a particular number. For example, the first level is level 0, the second level is level 1, and so on. In addition, a level can be a test level for execution by test satellites, a production level for execution for production satellites, or an obsolete level that is not executed by any satellite.

In the new copy of the setup and cleanup batches, you can only append new batch steps. You cannot modify the contents or the order of any existing batch steps. The next time the satellites synchronize, they will only execute the new batch steps of these two batches. All the satellites will execute the new batch steps in the same order, ensuring that the satellites remain consistent after they all execute the new batch steps. If you have new satellites that have never synchronized, they will execute all the batch steps, including the new ones. Because all the satellites in a group that have the same application version run the same setup and cleanup batch steps in the same order, they will have a similar database definition and data after they execute all the batch steps.

The new level of the update batch, however, can be modified in any way that you require. Unlike the situation that applies to setup and cleanup batches, a satellite executes all the batch steps of the update batch each time that it synchronizes. By definition, the update batch is idempotent. That is, a satellite can execute the update batch repeatedly without changing its current state or database definition.

The way that an application interfaces with its data does not change within a version of an application. Once you have set up the underlying database definition to support the application, it is unlikely that you will want to substantially modify it. Rather, you will want extend the database definition in small ways to address intermittent problems such as performance. For this reason, the setup and cleanup batches can only be appended to. Although the database definition sets the structure of the data, with a fixed number of tables and columns, the data content is subject to continual updates. One of the primary uses of the update batch is to synchronize data between the satellite and one or more data sources. You can change the data synchronization operation in any way to ensure that satellites have the data that they require. For this reason, the update batch in a new level is fully modifiable.

Suppose you have a situation in which the performance of the application is affected because the application is maintaining more data than was originally anticipated. In this situation, you would add a new batch step to the setup batch to create an index. All satellites in the group that have the same

application version will execute the new batch step to create the index when they next synchronize. Differences exist, however, in the number of setup batch steps that the satellite will execute when it next synchronizes, depending on when the satellite was rolled out:

- If a satellite was part of an earlier stage of the deployment, and has already executed all of the original group batches, this satellite will first execute the new batch step in the setup batch to create the indexes, then continue by executing the update batch. The new index will result in improved performance for the application.
- If a satellite is new and has not yet synchronized, the first time it synchronizes, the satellite will execute all the batch steps in the setup batch, including the new batch step that creates the indexes. When this satellite completes the synchronization process (that is, it executes all batch steps to the end of the cleanup batch), it will be consistent with older members of the group. This satellite will not report the performance impact on the application.

The levels of an application version allow you to maintain and manage the changes to the database definition within an application version. Because the Satellite Administration Center maintains a history of the different levels of the application version, you can track the changes that have occurred over the life time of the application version. You can view the changes that have happened from the Edit Application Version window.

Related concepts:

- “Test, Production, and Obsolete States of Application Version Levels” on page 208
- “States of an Application Version” on page 213
- “Life Cycle of an Application Version” on page 201

Related tasks:

- “Editing an application version : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Test, Production, and Obsolete States of Application Version Levels

Levels simplify the administration of groups because they allow you to test database definition changes on test satellites, without affecting your production satellites. Levels also allow you to maintain the hundreds (or thousands) of production satellites that belong to a group. Each level in a application version has one of the following states: test, production, or obsolete. When you create the first level for an application version, that level, and the batches associated with it, are in the test state. These test batches set up and maintain the database definition for the test satellites. You can modify

these test batches until you are satisfied with the results that they produce on the test satellites. Then, you promote the test level to production.

When you promote the level to production, the batches associated with this level can be executed by the production satellites. You will probably not want to enable all the satellites of the group to begin executing the production batches at the same time. Instead, consider rolling out subsets of the group satellites to execute the production batches. You can deploy subsets of the satellites according to a characteristic that the subset shares, such as a subgroup. When you enable these satellites, they will download and execute the production batches the first time that they synchronize. Over time, you will enable all the group satellites to execute the production batches.

When the production satellites are executing the batches of the production level, you may find it necessary to modify the database definition generated by the production batches to address a problem. For example, assume that performance of the application is becoming worse over time. You determine that adding indexes to a table will improve the performance of the application. To fix the problem, you create a new test level from the existing production level. You then append a batch step to the setup batch to create the indexes. Because the new level is in the test state, only the test satellites will be able to execute the new batch step of the test setup batch. When you are satisfied with the results that the test batch generates on the test satellites (that is, the performance of the application is again satisfactory), you promote the test level to the production level. The next time that they synchronize, the production satellites will download the new batch step and create the indexes.

The levels, and the associated states, represent the life cycle of the application version and its associated batches. Additional details about states are as follows:

Test You use a test level to try out database definition changes on the test satellites of your group. The batches of a test level can only be executed by test satellites. Whether you are deploying a new group, or you are testing changes to the database definition, you will want to test the batch steps in a test level to ensure that they produce the results that you want. You can only have one level in the test state. With only one test level, you always know which batches have been modified, and which changes are being tested.

The first level of an application version is always created in the test state. This level is level 0. When level 0 is created, all the batches and batch steps that you add to it are in the test state. When level 0 is in the test state, you can modify all of its batches and batch steps. This includes the reordering or deletion of batch steps, and the deletion of the batches.

You can also create a test level from the existing production level. When a new test level is created from a production level, the setup and cleanup batches can contain a mixture of production and appended test batch steps:

- The production steps of the setup and cleanup batches cannot be modified or reordered. Test batch steps can only be appended after the production batch steps to ensure that the changes that occur on the test satellites result from the test batch steps, and not from changes in the order of the production batch steps.
- The steps of the update batch in the test level are at the test state.

When you are satisfied with the results of a test level, you can promote it to production, so that the batches within it can be executed by production satellites.

Production

You use the production level to set up and maintain the database definition for the application version that your production satellites run. Because the batches associated with the production level are fully tested before being promoted, they provide predictable results when the satellites execute them. Using fully tested batches in a production level allows you to scale your satellite environment to any size that you require.

To ensure consistency among the satellites that execute them, production batches cannot be modified or deleted, nor can the batch steps within them. When a test level is promoted to production, all the batches associated with it are set to the production state.

You cannot directly create a production level of an application version. A production level is always a test level that was promoted to production. In this way, you can always use your test satellites to test and tune changes to batches. This helps to isolate testing from the production environment. You must always explicitly promote a test level to production before the production satellites can execute the new or changed batches.

You can only have one production level of an application version. If you have an existing production level and promote a test level to production, the existing production level becomes obsolete. The existing production level is obsoleted because the database definition that it sets up and maintains is no longer adequate to support the application.

If you find that one or more batches of a production level no longer meet all of your requirements, you can create a new test level from it.

Obsolete

An obsolete level is no longer adequate to support the application. For this reason, the batches of an obsolete level cannot be executed by any satellite. There are two ways in which a level can become obsolete:

- A new production level supersedes it. This occurs when a test level is promoted to production, and replaces an existing production level.
- You explicitly obsolete it when it is no longer required.

There can be many obsolete levels. If you want to be able to track the changes to an application version, you should keep them.

You cannot put an obsolete level back into production, nor can you create a new test level from an obsolete level.

The levels of an application version support a test/production/obsolete development model (or life cycle), which can be used in conjunction with the procedure that you use to implement your end-user application.

The following table is an overview of an application version, the available batch types, and the states at which a level can be:

Table 13. The Relationships among the Batches and Levels of an Application Version

Application Version			
Batches	Level is at Test State	Level is at Production State	Level is at Obsolete State
<p>At most, one of the following types of batches can be associated with any level of an application version:</p> <ul style="list-style-type: none"> • Setup • Update • Cleanup. 	<p>The batches of a test level can only be executed by test satellites.</p> <p>When a level is created, the level and the batches that it contains are always in the test state. The batch steps, however, may be a mixture of test and production batch steps. This enables you to validate the changes to the batches before making them available to the production satellites.</p> <p>An application version can contain only one test level.</p>	<p>The batches of the production level can only be executed by production satellites.</p> <p>The batches and batch steps of a production level cannot be modified in any way. This guarantees consistency among your production satellites.</p> <p>An application version can contain only one production level.</p>	<p>The batches of an obsolete level cannot be executed by any satellite. This occurs because obsolete batches are no longer adequate to support the end-user application.</p> <p>You can have multiple obsolete levels of an application version. If you keep the obsolete levels, you can use them to track the changes that have occurred to the database definition that supports the application.</p>

Related concepts:

- “Life Cycle of an Application Version” on page 201
- “Update Batch in a Test Level of an Application Version” on page 213

Related tasks:

- “Promoting a Test Level to a Production Level in an Application Version” on page 214
- “Creating a Test Level from a Production Level in an Application Version” on page 215
- “Obsoleting a Production Level in an Application Version” on page 216

States of an Application Version

Like levels, the application version also has a state. You can use the application version details view to determine the state of an application version. This view is available from the Satellite Administration Center. The state is displayed in the **State** column. The state of the application version is set as follows:

- If the application version does not have a production level, the state is not set. That is, no entry exists for the application version in the **State** column.
- If a production level exists for the application version, and none of the production satellites that execute the group batches of this application version are in the failed state, the state of the application version is **Production Normal**.

Note: The application version can also be in the **Production Normal** state if no production satellites are executing the batches associated with this application version.

- If a production level exists for the application version, and one or more of the production satellites that execute the group batches of this application version are in the failed state, the state of the application version is **Satellite Failed**.

Related tasks:

- “Viewing application version details : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Update Batch in a Test Level of an Application Version

When you create a new test level, the update batch steps that were copied from the production level of the update batch are changed to the test state. In this way, you can modify the steps and their order in the batch, as required. Because the update batch is, by definition, idempotent, it should not modify the state or database definition of the satellite. For example:

- If you use the update batch to back up the database, you may want to change the backup buffer size to obtain better performance.
- If an operation is timing out, you can change a parameter that controls the amount of time that it can take to complete.

In these examples, the changes are implemented by minor changes to existing batch steps in the update batch.

Assume that you have one mission-critical table on your production satellites, and that you want to back up this table each time that these satellites

synchronize. Assume also that you want a snapshot of the data of this table before the data is refreshed. You would create a test level from level 2, the current production level. Because you want the backup image to contain a snapshot of the data before it is refreshed, you would add the BACKUP TABLESPACE batch step before the batch steps that update the data:

Application Version				
Level	State	Setup Batch	Update Batch	Cleanup Batch
0	Obsolete	configure	synchronize data	
1	Obsolete	configure (with batch step added to create index)	synchronize data	
2	Production	configure (with batch step added to create index)	synchronize data	reorganize
3	Test	configure (with batch step added to create index)	synchronize data (BACKUP TABLESPACE followed by data synchronization)	reorganize

When satisfied with the results, you would promote level 3 to production. Level 2 would be automatically rendered obsolete.

Promoting a Test Level to a Production Level in an Application Version

When you create a level in an application version, it is created in the test state. By definition, only test satellites can execute the batches of a test level. After you are satisfied that the batches in the test level produce the results that you want, you will want the production satellites to execute them. To do this, you must promote the level to production.

Procedure:

To promote a test level, use the Edit Application Version window, which is available from the Satellite Administration Center.

If you already have a production level and you promote the test level, the existing production level will be moved to the obsolete state. This occurs because the previous production level is no longer adequate to support the end-user application. Because the test level is a copy of the batches and batch steps of the previous production level, no batch steps are lost from the previous production level. This means that the first time any new production satellites synchronize, they will run the same production batch steps in the

same order as satellites that were rolled out earlier. The resulting database definition on the new satellites will resemble that of other members of the group. In this way, consistency is maintained across the group.

Related concepts:

- “Life Cycle of an Application Version” on page 201

Related tasks:

- “Editing an application version : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Promoting a Batch to Production : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Creating a Test Level from a Production Level in an Application Version

You may find it necessary to refine or extend a setup, update, or cleanup batch that is associated with a production level. To do this, you can create a new test level from an existing production level.

Restrictions:

Existing batch steps in the setup and cleanup batches cannot be modified or reordered in a test level. Instead, you can only append test batch steps to these batches. You can modify the update batch in any way that you require.

Procedure:

Use the Edit Application Version window, which is available from the Satellite Administration Center, to add a test level from a production level. When you add a test level, all the batches and batch steps of the production level are copied to the test level.

In the case of the setup and cleanup batches, the test satellites will execute only the appended test batch steps. For the update batch, the test satellites will execute all the batch steps, regardless of whether the batch is changed. To modify the batches in a test level, use the Change Level notebook, which is also available from the Satellite Administration Center.

Related concepts:

- “Life Cycle of an Application Version” on page 201

Related tasks:

- “Changing the level of an application version : Satellite Administration Center help” in the *Help: Satellite Administration Center*

- “Editing an application version : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Obsoleting a Production Level in an Application Version

When you obsolete a production level, the batches associated with it can no longer be executed by any satellites. All the batch steps in the batches become obsolete.

Restrictions:

You can only obsolete a production level.

Procedure:

You obsolete a production level by using the Edit Application Version window, which is available from the Satellite Administration Center. You can obsolete a production level in two ways:

- By promoting a test level to a production level when there is already an existing production level. The existing production level is no longer adequate to support the end-user application, so it becomes an obsolete level.
- By selecting a production level and clicking the **Obsolete** push button. You obsolete a level to prevent the execution of its batches. You may, for example, be working on a new test level. Or, you have fully deployed a new version of the end-user application, which has a different application version. In this situation, you do not intend to create any new levels of the previous application version because it is no longer in production.

Related concepts:

- “Life Cycle of an Application Version” on page 201

Related tasks:

- “Promoting a Test Level to a Production Level in an Application Version” on page 214
- “Promoting the Batches of Test Level 0 to Production” on page 267
- “Editing an application version : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Relationships Between Batches and Batch Steps

Batches are ordered sets of batch steps. Batch steps, when in group batches, can be test, production, or obsolete batch steps, depending on the state of the associated level. Each batch step contains a script, as well as other information, which the satellites use to set up and maintain the database definition for the application that runs on the satellites.

The type of the batch step controls whether that batch step is executed by test or production satellites, and whether the batch step is modifiable. The relationships between the state of a level and the state of batch steps are as follows:

- When you first create an application version, the first level that you add to it is level 0. This level is at the test state, and is empty. It has no setup, update, or cleanup batches. As you create batches for level 0, their steps are in the test state. This occurs because you will want to rigorously test the batches and batch steps with your test satellites to ensure that they set up and maintain the database definition and data correctly.

Because all the batch steps are in the test state, you can modify or reorder them until you obtain the database definition and data that you require. If required, you can also delete a batch and replace it with a different batch. This characteristic of being completely modifiable only occurs with batches and batch steps associated with level 0 when it is in the test state.

- When you create a new test level from the existing production level, all the batches and batch steps in them are copied to the new test level. Within the new test level, the steps that were copied with the setup and cleanup batches are marked as production batch steps, and cannot be modified or reordered. You can, however, append additional test batch steps to these batches.

The copied production batch steps in the test level of the setup and cleanup batches cannot be modified, meaning that the changes to the database definition and data on the test satellites can only result from the new batch steps. The original batch steps are locked so that, if unexpected problems occur during the test phase, it will be much easier to identify and repair the problems that can occur. Changes cannot result from changes to the original batch steps, or from an interaction between changes in the original batch steps and the new batch steps.

You can modify the test batch steps until they provide the results that you want on your test satellites: you can even remove these steps. Because the test of the batch is isolated from the production satellites, normal production is not affected.

The batch steps that were copied from the update batch of the production level are marked as test batch steps in the new level of the update batch. You can modify these batch steps in any way that you require.

- A test level can be promoted to production. When the level is promoted to production, the batch steps of all batches are changed to the production state. Because the batch steps are in production, they are locked and cannot be modified. The batches and batch steps cannot be modified because you do not want untested changes introduced to the production satellites.

Ordinarily, only production satellites can execute production batch steps, and only test satellites can execute test batch steps. An exception occurs if a new test satellite is introduced to the group. In this situation, the new test satellite has not yet executed any batches. When this satellite first synchronizes, it will execute all the batch steps in the batches of a test level, both production and test steps. In addition, you can configure a test satellite to execute all the batch steps of a test batch, including any production batch steps in it.

- Obsolete batch steps only exist in the batches of an obsolete level. That is, obsolete batch steps cannot exist in either a test or a production level. Obsolete batch steps cannot be executed by any satellite.

Related concepts:

- “Update Batch in a Test Level of an Application Version” on page 213

Related tasks:

- “Promoting a Test Level to a Production Level in an Application Version” on page 214
- “Creating a Test Level from a Production Level in an Application Version” on page 215
- “Obsoleting a Production Level in an Application Version” on page 216
- “Creating an application version : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Storage of Script on a Satellite During a Synchronization Session

When a satellite synchronizes, it obtains the scripts that it is to execute from the satellite control server, then stores the scripts in a location that is specific to whether the script is for a setup, an update, or a cleanup batch. The execution results are also stored. When the synchronization session ends normally, the contents of these directories are deleted. If the synchronization session is interrupted (either the user stops the session or an abend occurs), the contents of these directories are not deleted.

Directory	Description
<i>instance_path</i> \satellite	The base directory for stored scripts Note: You should not modify the contents of this directory; otherwise, synchronization may not be able to occur.
<i>instance_path</i> \satellite\setup <i>instance_path</i> \satellite\setup\results	The directory where the scripts for a setup batch are stored. Results are stored in the \results directory.
<i>instance_path</i> \satellite\update <i>instance_path</i> \satellite\update\results	The directory where the scripts for an update batch are stored. Results are stored in the \results directory.
<i>instance_path</i> \satellite\cleanup <i>instance_path</i> \satellite\cleanup\results	The directory where the scripts for a cleanup batch are stored. Results are stored in the \results directory.

Fix Batches

You can use fix batches for a variety of situations:

- To fix problems on test satellites when a group batch in a test level does not produce the intended results.
- To fix problems that occur on production satellites.
- To fix problems that are caused by a fix batch. In this situation, you can either modify the original fix batch, or create a different fix batch that both undoes the results of the original fix batch and applies a different fix.

Note: If a specific fix batch is assigned to more than one satellite, you should not modify the batch until you are sure that all the satellites that are assigned to execute it have done so. If the fix is unsatisfactory, it will be unsatisfactory across all the satellites that execute it, and you will have a consistent starting point from which to apply a different fix.

- To query the results of a previous fix batch.
- To make a change to a satellite that is not required by any other satellite.
- To query the current state of a satellite.

Because the batch steps are not locked, you can repeatedly modify the steps in a fix batch until the problem on the satellite is fixed to your satisfaction. When you are satisfied with the results of a fix batch on a satellite, you promote that satellite back to executing its group batches.

You work with fix batches by using the Create Batch and Edit Batch windows.

Related tasks:

- “Identifying and Fixing a Failed Satellite” on page 337
- “Creating a batch : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Editing a batch : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Viewing which satellites are using a fix batch : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Promoting a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Chapter 8. Authentication in the Satellite Environment

Authentication Credentials	221	Authentication with Target Servers for Script Execution	224
Authentication Credentials Stored at the Satellite Control Server	222	Password Change Management	224
Storage of Authentication Credentials on Satellites	223	Managing Password Changes for Access to the Satellite Control Server	225
Creation and Maintenance of Authentication Credentials on a Satellite	223	Managing Password Changes at Target DB2 Servers	226

In the satellite environment, the administration solution is based on the synchronization between satellites and the satellite control server. For synchronization to occur, you must set up a variety of authentication credentials, both to enable the satellite to download the scripts, and to enable the satellite to execute the scripts.

Authentication Credentials

An authentication credential is the combination of a user ID and a password. Almost every activity in the satellite environment requires authentication, from the first connection to the satellite control database for a test synchronization, to the execution of a script. Authentication credentials reside both at the satellite control server (in the satellite control database), and at every satellite in the environment. The master copy is at the satellite control server. Every satellite maintains a shadow copy of the authentication credentials. Because all passwords in the satellite environment are encrypted, the authentication information on the satellites cannot be updated independently of the synchronization process with the satellite control server. The encryption prevents unauthorized access to a password for a user ID, and allows you to maintain tight control over the security of the environment.

Related concepts:

- “Authentication Credentials Stored at the Satellite Control Server” on page 222
- “Storage of Authentication Credentials on Satellites” on page 223

Related tasks:

- “Creating an authentication : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Authentication Credentials Stored at the Satellite Control Server

The satellite control server maintains the master copy of all the authentication credentials that are required in the satellite environment. Because all authentication credentials are at the satellite control server, you can manage all your authentication credentials from a central location.

When you create an authentication credential, you give it a name and provide a user ID and password. The password is encrypted when it is stored in the satellite control database. When you create a target, you provide the name of the authentication credential. The user ID and password of this authentication credential is used to authenticate the user with the target.

You must create an authentication credential for every DB2[®] instance or DB2 database that will be the target of script execution. Because all the satellites in a group execute the same scripts against the same targets, you only need to create one set of authentication credentials for the group. Because operating system scripts run locally on the satellite under the authority of the system administrator, you do not need to create an authentication credential for them.

Because you will know which targets the satellites in a group will need to authenticate with before you set up a model office or perform a deployment, you may find it more convenient to create the authentication credentials and targets before creating any batches. You can set up and maintain the authentication credentials by using the Create Authentication and Edit Authentication windows. You can set up and maintain targets by using the Create Target and Edit Target windows. These windows are available from the Satellite Administration Center.

Note: The maximum supported length for a password is 31 characters.

Related concepts:

- “Components of a Batch Step” on page 188

Related tasks:

- “Managing Password Changes for Access to the Satellite Control Server” on page 225
- “Managing Password Changes at Target DB2 Servers” on page 226
- “Creating an authentication : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Editing an authentication : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Creating a target : Satellite Administration Center help” in the *Help: Satellite Administration Center*

- “Editing a target : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Storage of Authentication Credentials on Satellites

All user IDs and passwords that are used in the satellite environment for synchronization are stored in the `instance_directory\security\satadmin.aut` file on each satellite. The entries in this file mirror the authentication credentials at the satellite control server. The satellite synchronization process maintains the `satadmin.aut` file by downloading the encrypted passwords from the satellite control database, and storing the encrypted passwords in the file. The use of encryption prevents direct access to all passwords that are used in the satellite environment.

Only satellites that are both recorded in the satellite control database and enabled to execute group batches can use the synchronization process to access and download the authentication credentials.

During a synchronization session, two types of security checks occur. First, the satellite must authenticate with the satellite control server to download the batch steps that it will execute. Second, when each script is executed, if its target is an instance or a database, authentication occurs before the script can be executed. If the target of the script is the local operating system, the script executes locally on the satellite under the authority of the system administrator.

Related concepts:

- “Authentication with Target Servers for Script Execution” on page 224

Related tasks:

- “Creating the User ID and Authentication Credential Required for Satellite Synchronization” on page 242
- “Recreating or Updating the `satadmin.aut` File on a Satellite” on page 346

Creation and Maintenance of Authentication Credentials on a Satellite

The `satadmin.aut` file is created either during installation, or at the first test synchronization. During the installation process, you can supply a user ID and password that will be used to connect to the satellite control database on the satellite control server. If you supply this information during installation, the authentication file is created and the user ID and password stored in it. If you do not provide the user ID and password during installation, you will be

prompted for them the first time you run the **db2sync -t** command to perform a synchronization test. At this time the file will be created and the authentication credentials stored in it.

The first time that the satellite synchronizes, it downloads the authentication credentials for all the targets against which it must authenticate, and stores this information in its authentication file. All the passwords in the file will be encrypted. On subsequent synchronizations, changes to the authentication credentials are also downloaded so that the authentication credentials on the satellite are always current.

Related tasks:

- “Managing Password Changes for Access to the Satellite Control Server” on page 225

Related reference:

- “db2sync - Start DB2 Synchronizer Command” in the *Command Reference*

Authentication with Target Servers for Script Execution

When a satellite executes a script against a target that is a DB2® instance or a DB2 database, the user ID associated with the authentication credentials must be authenticated before the script can be executed. Authentication credentials, the combination of a user ID and password, are required to authenticate on each attach to a DB2 instance, or connect to a DB2 database. When the satellite executes the batch step, its authentication file is accessed to retrieve the user ID and password for the target, which is then used on the attach or connect to authenticate the satellite.

Password Change Management

Because of standard security procedures, situations will occur in which the password must be changed for one or more of the target DB2® servers against which satellites authenticate. When this occurs, however, the satellites may experience authentication errors when they attempt to attach or connect to the target DB2 server. To avoid this problem, the satellite maintains both the current and the previous password for all of the DB2 databases or servers against which that satellite must authenticate. The fact that the satellite maintains both passwords enables you to make a transition from the existing password on a DB2 server to the new password.

Related tasks:

- “Managing Password Changes for Access to the Satellite Control Server” on page 225

- “Managing Password Changes at Target DB2 Servers” on page 226

Managing Password Changes for Access to the Satellite Control Server

To maintain security, you will need to periodically change the password that satellites use to synchronize with the satellite control server.

Procedure:

To change the password associated with the user ID that one or more groups use to synchronize with the satellite control server:

1. Identify the authentication credentials that are being used by the group. You can use the Edit Group window to see the name of the authentication credentials that are being used by the group. This window is available from the Satellite Administration Center.
2. Edit the named authentication credentials to change the password associated with the user ID. To perform this task, use the Edit Authentication window.

When you change the password used to access the satellite control server, the **Password changed** column of the satellite details view in the Satellite Administration Center changes to Yes for all the satellites in the group. The Yes value indicates that the password required to access the satellite control server is changed.

When a satellite next synchronizes, it will recognize that the password is changed, and download the changed password. As soon as the satellite obtains the new password, the **Password changed** column of the satellite details view in the Satellite Administration Center changes to No to indicate that the satellite has the new password.

The next time that the satellite synchronizes, it will use the new password to connect to the satellite control server. Because the password is not yet changed at the satellite control server, authentication will fail. When the authentication fails, the satellite will attempt to connect again, this time using the old password. This time, the authentication will succeed.

3. When all the satellites have updated their authentication file with the new password (that is, the **Password changed** column of the satellite details view in the Satellite Administration Center is No for all the satellites of the group), change the password that is required to access the satellite control server. You can use the ATTACH command with the CHANGE PASSWORD parameter to change the password for the satellite control server, or you can use the operating system security manager.

Related tasks:

- “Editing an authentication : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Editing a group : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Viewing satellite details : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Managing Password Changes at Target DB2 Servers

To maintain security, you will periodically need to change the password at one or more target DB2 servers that a satellite will access when it synchronizes.

Procedure:

To change the password at a target DB2 server:

1. Edit the execution target to determine which authentication credential it is using. To perform this task, use the Edit Target window in the Satellite Administration Center.
2. Edit the authentication credentials to change the password that the satellites require to be authenticated at the target DB2 server to the new password that you intend to use on the target server. To perform this task, use the Edit Authentication window in the Satellite Administration Center.
3. Change the password on the target DB2 servers using the facilities provided by the operating system.

When the satellite next synchronizes, any new passwords are downloaded and stored, in encrypted format, in the `satadmin.aut` file on the satellite. The satellite will use the new password when it attempts to connect to the target DB2 server. If the password is not yet changed at the target server, authentication will fail. The satellite automatically recovers from this error by attempting to connect or attach again, this time using the old password. Authentication should succeed.

An authentication error can occur if a synchronization session is stopped or terminates before the satellite has executed all the scripts for that session. When an interrupted synchronization session is restarted, the satellite will execute the remaining scripts for the session before connecting to the satellite control server to report the results of the session. The error occurs if any scripts that remain to be executed access a target DB2 server whose password changed between the time that the synchronization session was stopped and restarted. Because the satellite does not download updated passwords until it connects to the satellite control server, the satellite will not have the correct

password for the DB2 target that has undergone a password change. You can recover from this situation by refreshing the authentication credentials on the satellite.

Related tasks:

- “Recreating or Updating the satadmin.aut File on a Satellite” on page 346
- “Editing an authentication : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Editing a target : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Chapter 9. Cataloging Instances and Databases

Requirement to Catalog Instances and Databases in the Control Center Instance	229	Cataloging Local Instances on the Model Office	234
Requirement to Catalog Systems, Instances, and Databases on the Control Center	230	Completing the Setup of the Model Office	235
Cataloging the Satellite Control Server and the Satellite Control Database on the Control Center	231	Cataloging Instances and Databases on Test Satellites	236
Cataloging the Model Office on the Control Center	231	Setting up a Test Satellite Using a Client Profile	237
Cataloging Remote Instances and Databases on the Model Office	232	Cataloging Instances and Databases on Production Satellites	237
Cataloging Remote Instances and Databases on the Model Office Using a Customized Client Profile (Windows)	233		

To be able to use the satellite administration solution, you must catalog the instances and databases on each system, including the satellites, in the configuration. The sections that follow describe what DB2 objects must be cataloged on each system, and suggest techniques that you can use to catalog the required entries.

Requirement to Catalog Instances and Databases in the Control Center Instance

The Control Center and the Satellite Administration Center require access to the instance of the satellite control server, and the satellite control database. If the satellite control server instance and the satellite control database do not show up in the Control Center, you need to catalog them in the instance from which the Control Center is started. You can catalog them using the Control Center.

Note: The Control Center instance is the instance where the DB2[®] JDBC server is running. The Control Center uses the DB2 JDBC server instance. The Windows[®] Services refer to server as the DB2 JDBC Applet Server.

When you have cataloged the satellite control server and the satellite control database, you can use the Satellite Administration Center to create groups and satellites, and to define batches to be executed by the satellites.

A batch step executes against an execution target, which can be a DB2 instance, a DB2 database, or the operating system on the satellite. You must create a named target for every DB2 instance or DB2 database against which a

batch step will execute. You do not need to create a named target for the batch steps that execute against an operating system.

Before you can create a named target for a DB2 instance or DB2 database using the Satellite Administration Center, you must first catalog the following targets in the node and database directories of the Control Center instance.

- The instances and databases of the model office, as these are the targets of the administrative scripts.

An additional advantage to cataloging the model office instances and databases in the Control Center instance is that you can then use the Control Center both to examine the model office, and to generate scripts to manipulate the model office. To create the scripts, use the Show SQL and Show Command facilities of the Control Center windows and notebooks. You can also use the Control Center to manage the model office, for example, to back up the databases on it.

If you want your help desk to use the Control Center and the Satellite Administration Center to perform problem determination and diagnosis on satellites, you should catalog the satellite control server and the satellite control database in the Control Center instance that the help desk will use.

Related tasks:

- “Cataloging the Satellite Control Server and the Satellite Control Database on the Control Center” on page 231

Requirement to Catalog Systems, Instances, and Databases on the Control Center

You can use the Control Center to catalog the required systems, instances and databases. First, catalog the satellite control server and the satellite control database. Then catalog the model office.

Note: When you catalog systems and instances, you *must* specify TCP/IP as the protocol to use for communications. When you export the connection information from the Control Center, then import this information to the model office, all of its system and instance node directory entries will specify TCP/IP, which is the only protocol supported in a satellite environment.

Related tasks:

- “Cataloging the Satellite Control Server and the Satellite Control Database on the Control Center” on page 231
- “Cataloging the Model Office on the Control Center” on page 231

Cataloging the Satellite Control Server and the Satellite Control Database on the Control Center

Before you can create a satellite environment, you must catalog the satellite control server and the satellite control database on the Control Center.

Note: Examine the object tree of the Control Center under the system where the satellite control server is running. If the DB2CTLSV instance and the SATCTLDB database already appear, you do not have to perform these steps. This situation occurs when the Control Center's JDBC server is using the satellite control server instance.

Restrictions:

TCP/IP is the only supported communications protocol in a satellite environment.

Procedure:

To catalog the satellite control server and the satellite control database in the Control Center object tree, use the Add System, Add Instance, and Add Database windows, which are available from the Control Center.

Related concepts:

- "Discovery of administration servers, instances, and databases" in the *Administration Guide: Implementation*

Related tasks:

- "Adding a database: Control Center help" in the *Help: Control Center*
- "Adding an instance: Control Center help" in the *Help: Control Center*
- "Adding a system: Control Center help" in the *Help: Control Center*

Cataloging the Model Office on the Control Center

You must catalog the model office on the Control Center so that you can administer the model office using DB2 GUI tools.

Restrictions:

TCP/IP is the only supported communications protocol in a satellite environment. Also, when you specify the operating system on the model office, you must select Windows.

Procedure:

To catalog a model office and its instances and databases in the Control Center, use the Add System, Add Instance, and Add Database windows, which are available from the Control Center.

Related tasks:

- “Adding a database: Control Center help” in the *Help: Control Center*
- “Adding an instance: Control Center help” in the *Help: Control Center*
- “Adding a system: Control Center help” in the *Help: Control Center*

Cataloging Remote Instances and Databases on the Model Office

When a satellite synchronizes, it connects to the satellite control database at the satellite control server. To connect (and consequently to synchronize), the satellite must have catalog entries for the satellite control server instance and the satellite control database in its node and database directories.

When the satellite downloads scripts from the satellite control server to execute, the DB2 instance or database that is the execution target must already be cataloged on the satellite.

Restrictions:

TCP/IP is the only supported communications protocol in a satellite environment.

Procedure:

You can use two methods to catalog the instances and databases that are remote to the model office:

- Issue the CATALOG TCPIP NOTE and CATALOG DATABASE commands through the CLP to catalog the remote instances (nodes) and databases.

If you use the CATALOG commands, you must ensure that the instance and database alias names that you specify for the command match those that are recorded in the instance of the Control Center. These names are the same names that you use when you create execution targets in the Satellite Administration Center. One simple way to ensure that the names recorded at the model office match those recorded at the Control Center is to write a script that issues the CATALOG commands. You then run the script on both the Control Center instance and the model office instance. As an alternative, you can also use the **Show Command** button on the different Control Center windows that you use to catalog instances and databases to display the CATALOG command, then save the command to a file which you can later execute on the model office.

- You can export a client profile to a file from the instance of the Control Center. You then import this file on the model office. The file contains the information that is required to create the node and database directory entries on the model office.

Importing the client profile from the Control Center provides one major advantage: you ensure that the instance and database alias names are identical to those in the instance of the Control Center. You use these names when you use the Satellite Administration Center to create execution targets.

You may have to catalog other nodes and databases on the model office, for example, if your application will be using the satellite as a client to access another DB2 system. You can create entries in the node and database directories on the model office by using the CATALOG TCPIP NODE and the CATALOG DATABASE commands.

Related tasks:

- “Cataloging Remote Instances and Databases on the Model Office Using a Customized Client Profile (Windows)” on page 233

Related reference:

- “CATALOG DATABASE Command” in the *Command Reference*
- “CATALOG TCP/IP NODE Command” in the *Command Reference*

Cataloging Remote Instances and Databases on the Model Office Using a Customized Client Profile (Windows)

You can use a customized client profile to catalog remote instances and databases on the model office.

Restrictions:

The procedure that follows applies to Windows environments only.

Procedure:

To use a customized client profile to perform cataloging on the model office, perform the following steps.

1. On the same system where you are running the Control Center, use the Configuration Assistant to create the customized client profile file.

Notes:

- a. When you create the customized client profile file, do not select the database that is on the model office.

- b. When you create the customized client profile file, ensure that **Client settings** and **CLI/ODBC common settings** are deselected.
 - c. If you are running the Control Center on the same system as the satellite control server and the satellite control database, do not select the SATCTLDB database. If you select this database, it will be set up on the model office as a local database.
2. On the model office:
 - a. Open a command window.
 - b. Issue the **db2cfimp** *file_name* command to import the customized client profile file.

Note: If you are running the Control Center on the same system as the satellite control server and the satellite control database, and you did not select the satellite control database when you created the customized client profile file, you will have to catalog the satellite control server instance, DB2CTLSV, and the SATCTLDB database on the model office using the CATALOG TCPIP NODE and CATALOG DATABASE commands.

Related tasks:

- “Exporting a customized configuration profile : Configuration Assistant help” in the *Help: Configuration Assistant*

Related reference:

- “CATALOG DATABASE Command” in the *Command Reference*
- “CATALOG TCP/IP NODE Command” in the *Command Reference*
- “db2cfimp - Connectivity Configuration Import Tool Command” in the *Command Reference*

Cataloging Local Instances on the Model Office

If any local instance on the model office will be used as an execution target, you must catalog this instance on the model office using the same name that you used when you cataloged the instance on the Control Center.

Procedure:

To catalog local instances on the model office, map the instance name used on the Control Center to the actual local instance name using the CATALOG LOCAL NODE command.

For example, assume that the name that was given to the local instance when it was created is DB2 (On Windows-based platforms, DB2 is the default

instance that is created by the installation program). Also assume that you called the DB2 instance on the model office FINANINS when you cataloged it on the Control Center. Because FINANINS is the execution target of scripts that you want to run against the local instance called DB2, you must run the following command on the model office:

```
CATALOG LOCAL NODE FINANINS INSTANCE DB2
```

This command makes FINANINS an alias for DB2 on the model office.

Related reference:

- “CATALOG LOCAL NODE Command” in the *Command Reference*

Completing the Setup of the Model Office

As you begin to use the model office during your development phase, you need to configure it to resemble a production environment.

Procedure:

To finish configuring the model office so that it resembles a production environment:

1. Optional. Set up the ODBC/CLI values that you need.
2. Optional. Modify database manager configuration values with the UPDATE DATABASE MANAGER CONFIGURATION command.
3. Optional. Modify database configuration values with the UPDATE DATABASE CONFIGURATION command.
4. Optional. Modify registry variable values with the **db2set** command.

When your model office is configured to represent the production environment, you can transfer its configuration values (in addition to the cataloged entries for instances and databases cataloged on it), to your test and production satellites by using the **db2cfexp** and **db2cfimp** commands.

Related reference:

- “UPDATE DATABASE CONFIGURATION Command” in the *Command Reference*
- “UPDATE DATABASE MANAGER CONFIGURATION Command” in the *Command Reference*
- “db2set - DB2 Profile Registry Command” in the *Command Reference*
- “db2cfimp - Connectivity Configuration Import Tool Command” in the *Command Reference*
- “db2cfexp - Connectivity Configuration Export Tool Command” in the *Command Reference*

Cataloging Instances and Databases on Test Satellites

A test satellite is a copy of the model office that you use for testing scripts and replication. You can use test satellites during the development process, or, after you deploy the production satellites, to test scripts that will change the database definition.

Procedure:

So that test satellites can execute batches, catalog all the execution targets on the test satellite. These instances and databases must be cataloged on the test satellite under the same instance and database alias names that are used on the model office. The simplest way to achieve this is to export a client profile file from the model office using the **db2cfexp** command, then import the file on each test satellite using the **db2cfimp** command.

In addition, if you are using the test satellites to verify the application that you intend to deploy on the production satellites, consider setting up ODBC/CLI for each database, and configuring the database, database manager configuration and DB2 registry information so that you can test the application in an environment that is very similar to the production environment. If you have set up this information on the model office, you can use the **db2cfexp** and **db2cfimp** commands to copy this information from the model office to the test satellite rather than having to use the UPDATE DATABASE CONFIGURATION, UPDATE DATABASE MANAGER CONFIGURATION, and **db2set** commands to update the configuration of the test satellites

Related tasks:

- “Setting up a Test Satellite Using a Client Profile” on page 237

Related reference:

- “UPDATE DATABASE CONFIGURATION Command” in the *Command Reference*
- “UPDATE DATABASE MANAGER CONFIGURATION Command” in the *Command Reference*
- “db2set - DB2 Profile Registry Command” in the *Command Reference*
- “db2cfimp - Connectivity Configuration Import Tool Command” in the *Command Reference*
- “db2cfexp - Connectivity Configuration Export Tool Command” in the *Command Reference*

Setting up a Test Satellite Using a Client Profile

You can automate the configuration of the test satellite during the development process by exporting a client profile from the model office and importing it on the test satellite.

Procedure:

Use the **db2cfexp** command with the **TEMPLATE** option to generate a client profile from the model office. When DB2 is installed on the test satellite, import the client profile by using the **db2cfimp** command.

When you begin to deploy your production satellites, you can use the same method that you use to deploy production satellites both to create the required catalog entries on the test satellites, and to configure the test satellites to have the same configuration as the production satellites. You can then use the test satellites to test proposed changes to the production satellites.

Related tasks:

- “Performing a Mass Deployment” on page 309

Related reference:

- “db2cfimp - Connectivity Configuration Import Tool Command” in the *Command Reference*
- “db2cfexp - Connectivity Configuration Export Tool Command” in the *Command Reference*

Cataloging Instances and Databases on Production Satellites

Assuming that you maintain the model office throughout the development cycle so that it truly represents your production satellites, you can use the node and database directory entries, ODBC settings, database manager configuration values and DB2 registry values of the model office to set up your production satellites.

Procedure:

To automate the set up of the production satellites, export a client profile from the model office and import it to the production satellite. Use the **db2cfexp** command with the **TEMPLATE** option to generate a client profile from the model office. When DB2 is installed on the satellite, import the client profile to the production satellite by using the **db2cfimp** command.

To automate the process, you can export the client profile from the model office when you run the response file generator utility, **db2rspgn**, to generate the response file that you will use to install your production satellites. The install program will import the client profile to set up the production satellites.

Related tasks:

- “Performing a Mass Deployment” on page 309

Related reference:

- “db2cfimp - Connectivity Configuration Import Tool Command” in the *Command Reference*
- “db2cfexp - Connectivity Configuration Export Tool Command” in the *Command Reference*
- “db2rspgn - Response file generator” on page 124

Chapter 10. Setting up and Testing Your Satellite Environment

Setting up and Testing a Satellite Environment	239	Creating and Testing Group Batches	254
Preparing for a Test Synchronization	240	Creating Authentication Credentials.	255
Creating the User ID and Authentication Credential Required for Satellite Synchronization	242	Creating Execution Targets.	256
Granting Privileges on the Satellite Control Database.	243	Creating an Application Version for a Group	257
Recommended Privileges to Grant on the Satellite Control Database	245	Creating Level 0 of an Application Version	258
Recommended Privileges to Grant on Stored Procedures and Bind Files	247	Editing Level 0 of an Application Version to Create or Modify Group Batches.	259
Creating a Group for the Satellites	247	Changing Batch Steps in a Group Batch	260
Creating Test Satellites in the Satellite Administration Center	248	Testing Group Batches	262
Creating the Satellite Authentication File on a Satellite	249	Enabling Test Satellites to Execute Test-Level Batches	262
Setting the Application Version on a Satellite	249	Synchronizing Test Satellites to Execute Test-Level Batches	263
Setting the DB2SATELLITEID Registry Variable on a Satellite	250	Checking the Results of the Synchronization Session	264
Verifying the Setup Before the Test Synchronization	252	Fixing Problems Caused by Test-Level Group Batches	265
Testing the Synchronization Capability of a Satellite	253	Promoting the Batches of Test Level 0 to Production	267
		Setting the Execution Starting Point for a Satellite	267

The sections that follow describe how to set up and test a satellite environment

Setting up and Testing a Satellite Environment

Before performing a mass deployment of production satellites, you must set up and test the satellite environment. Setting up and testing the satellite environment consists of a variety of different tasks. Some of these tasks you perform using the Satellite Administration Center, while others occur at the satellite.

Procedure:

To set up and test a satellite environment, perform the following steps:

1. Prepare for a test synchronization.
2. Create and test the group batches.

Related tasks:

- “Setting up and Testing a Satellite Environment” on page 239
- “Creating and Testing Group Batches” on page 254
- “Performing a Mass Deployment” on page 309

Preparing for a Test Synchronization

You run a test synchronization to ensure that a satellite can connect to, and is recognized by, the satellite control server. Before a test synchronization can be done, a number of different pieces of information must be created on both the satellite control server, and on the satellite from which the test synchronization will be executed.

Prerequisites:

Before you can use the Satellite Administration Center, you must ensure that a satellite control database is cataloged on the Control Center.

The Satellite Administration Center is enabled when the Control Center detects that any of the cataloged instances contain a satellite control database. You can open the Satellite Administration Center from the pop-up menu associated with the instance that contains the satellite control database, or from the pop-up menu associated with the satellite control database. You can also open it from the Control Center tool bar, or the Tools menu in the tool bar.

Note: You should only administer the satellite environment from the Satellite Administration Center.

Procedure:

To prepare for a synchronization test, perform the following steps:

1. Prepare the satellite control server:
 - a. Install the satellite control server. DB2 Enterprise Server Edition can function as the satellite control server on either of the following:
 - Windows-based platform. When you select the Satellite Control Server component when installing DB2, the satellite control server and the satellite control database are automatically created. If you want to, you can customize the satellite control database.
 - AIX platform. On AIX, the satellite control server and the satellite control database are not automatically created when you install DB2 with the Satellite Control Server component.
 - b. Create the user ID and password, and the authentication credentials that the satellites require to synchronize.

- c. Grant access to the satellite control database for this user ID. You can also grant administrative and help-desk personnel access to the satellite control database.
 You should also ensure that the satellites have the required privileges on the stored procedure and bind file that are required for synchronization.
 - d. Create a group.
 - e. Create one or more test satellites for the group.
2. Install and prepare the test satellite for the test synchronization.
 Before a satellite can connect to the satellite control server, three different sets of information must be set up on the satellite: the authorization file, the application version, and the satellite ID.
 - a. Install DB2 on the satellite. On Windows-based platforms, DB2 Personal Edition, DB2 Workgroup Server Edition, and DB2 Enterprise Server Edition can all function as satellites.
 - b. Set up the satellite authentication file on the satellite.
 - c. Set the application version on the satellite.
 - d. Set the satellite ID on the satellite.
 3. Run the test synchronization.

Related tasks:

- “Customizing the SATCTLDB Database on Windows” on page 9
- “Setting Up the Satellite Control Server on AIX” on page 7
- “Creating the User ID and Authentication Credential Required for Satellite Synchronization” on page 242
- “Granting Privileges on the Satellite Control Database” on page 243
- “Creating a Group for the Satellites” on page 247
- “Creating Test Satellites in the Satellite Administration Center” on page 248
- “Creating the Satellite Authentication File on a Satellite” on page 249
- “Setting the Application Version on a Satellite” on page 249
- “Setting the DB2SATELLITEID Registry Variable on a Satellite” on page 250
- “Testing the Synchronization Capability of a Satellite” on page 253
- “Cataloging the Satellite Control Server and the Satellite Control Database on the Control Center” on page 231

Creating the User ID and Authentication Credential Required for Satellite Synchronization

In the satellite environment, you administer groups of satellites, and not individual satellites. This basic principle also applies to the authentication that is required to access the satellite control server and the satellite control database. All the satellites of the group use a group-level user ID and password when they connect to the satellite control database for synchronization.

Procedure:

To create the user ID and authentication credential that are required for synchronization:

1. Decide which user ID and password that you want the group of satellites to use when they connect to the satellite control database for synchronization.
2. Define this user ID and password to the operating system security manager using the tools provided by the operating system on which the satellite control server resides. The user ID and password must be set up at the operating system level because authentication is performed by the operating system.
3. Create an authentication credential using the Create Authentication window of the Satellite Administration Center.

Ensure that the authentication credential has the same user ID and password that you just specified at the operating system. When you create the authentication credential, give it a name that is meaningful. In one of the steps that follows, you will create a group of satellites. When you create the group, you will specify that this authentication name be used by the group.

Notes:

1. You can use a group ID instead of a user ID.
2. User IDs and passwords are case sensitive.

You specify this same user ID and password either when you install DB2 on a satellite that will belong to the group, or when you run a synchronization test for the first time from the satellite. The satellite requires the user ID and password so that it can authenticate with the satellite control server. Each satellite that belongs to the group will use the same user ID and password to authenticate with the satellite control server for the purpose of synchronization.

After you create the authentication credentials required for synchronization, you grant the synchronizing user ID access to the satellite control database.

Related tasks:

- “Granting Privileges on the Satellite Control Database” on page 243
- “Creating an authentication : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Granting Privileges on the Satellite Control Database

After you create the user ID that satellites will use to synchronize, you need to grant table privileges both to this user ID, and to administrative and help-desk staff.

You use standard DB2 authentication and authorization mechanisms to control access to the satellite control database. For more information about how to grant privileges to tables, refer to the topic on the GRANT statement. You can also use the Privileges notebook in the Control Center to grant privileges on the tables.

Procedure:

To grant access to the satellite control database both to satellites, and to administrative and other support staff:

1. Grant access to the group satellites.

Each group of satellites uses one authentication credential to both connect to the satellite control server, and to access the tables of the satellite control database. You specify the name of this authentication credential when you create the group. (After the group is created, you can use the Edit Group window to see the name of the authentication credential that the group satellites use to access the satellite control server. You can use the Edit Authentication window to obtain the user ID that is associated with the authentication credential.)

You must grant appropriate table privileges to the user ID of the group-level authentication credential so that the group satellites can access the tables in the satellite control database when they synchronize. Recommended Privileges to Grant on the Satellite Control Database lists the recommended privileges that you should grant to this user ID for the satellite control tables. To simplify the administration of the privileges, you can create a group at the operating system level to contain the user ID associated with the authentication credential for each group. If you use an operating system group, you can grant and manage database and table

privileges for the operating system group, instead of having to manage the privileges for each user ID for each group of satellites. To set up and manage the privileges:

- a. Create a group in the operating system security manager for the user IDs.
- b. Grant the required authorities to the group.
- c. Add additional user IDs to this group as you add new groups to the satellite environment, as required.

If you change the user ID that the group satellites use to access the satellite control server and the satellite control tables, add it to the operating system group to ensure that this user ID has the appropriate table privileges.

If you change the password that is associated with the user ID that the group satellites use to synchronize, the user ID's table privileges are not affected.

2. Grant access to the administrative and help desk staff.

In your organization, you may have staff who require administrative authority to the satellite control database, SATCTLDB. If you have a help desk, likely the help desk staff also require access to the satellite control database. You should consider using two operating system groups to control access to the satellite control database, one for administrative staff, and one for help desk staff. If you use groups, you can grant and manage database and table privileges for two groups, instead of having to manage the privileges of each user separately. To set up and manage the privileges for the administrative staff:

- a. Create a group in the operating system security manager for the satellite administrators.
- b. Grant the required authorities to the group (DBADM authority, for example, on the SATCTLDB database).
- c. Add additional administrative staff to this group, as required.

You can create a group for your help-desk staff following a similar procedure. Recommended Privileges to Grant on the Satellite Control Database lists the minimum recommended authorities that you should grant to the group to which the help desk staff belong. Depending on your requirements, the recommendations may be too restrictive, as they only enable the help-desk staff to view information when they use the Satellite Administration Center. For example, if you want help desk staff to be able to use the Satellite Administration Center to fix satellites, the group for the help desk staff requires the following table privileges:

- To apply an existing fix batch to a satellite, grant the UPDATE privilege on the SATELLITES table.

- To enable or disable a satellite, grant the UPDATE privilege on the SATELLITES table.
- To modify the execution starting point for a satellite, grant the UPDATE privilege on the SATELLITES table.

You may want the help desk staff to use the Satellite Administration Center to perform other tasks as well. For example:

- To reset passwords, grant the UPDATE privilege on the TARGET_AUTH table, and the EXEC privilege on the satencrypt user-defined function.
- To delete log records, grant the DELETE privilege on the LOG table.
- To change success code sets, grant the INSERT, UPDATE, and DELETE privileges on the following tables:
 - SUCCESS_CODES
 - SUCCESS_RELATIONS.

Related concepts:

- “Security” in the *Administration Guide: Planning*

Related tasks:

- “Creating a group : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Editing an authentication : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Editing a group : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Granting and revoking table privileges for users : Control Center help” in the *Help: Control Center*
- “Granting and revoking table privileges for groups : Control Center help” in the *Help: Control Center*

Related reference:

- “GRANT (Database Authorities) statement” in the *SQL Reference, Volume 2*
- “GRANT (Table Space Privileges) statement” in the *SQL Reference, Volume 2*

Recommended Privileges to Grant on the Satellite Control Database

In the following table, recommendations for privileges that should be granted to the tables in the satellite control database are provided. Before granting these privileges, you must grant the CONNECT privilege on the SATCTLDB database to:

- The group that contains the user ID associated with the synchronization authentication credential for each group
- The group to which your administrative staff belong
- The group to which your help-desk staff belong.

Note: A blank cell in the following table indicates that no privilege is required for the table in the satellite control database.

Table Name	Privilege for Group Authentication Credential	Privilege for Help-Desk Staff Group
TARGET_AUTH	SELECT	SELECT
TARGETS	SELECT	SELECT
GROUPS		SELECT
SCRIPTS	SELECT	SELECT
SUCCESS_CODES	SELECT	SELECT
SUCCESS_RELATIONS	SELECT	SELECT
BATCHES	SELECT	SELECT
BATCH_STEPS	SELECT	SELECT
APP_VERSIONS		SELECT
GROUP_BATCHES	SELECT	SELECT
SATELLITES	SELECT, UPDATE	SELECT
LOG	INSERT	SELECT

Note: When you create the authentication credential that the satellites use to synchronize in test mode, you can set up the user ID to have minimal access to the tables in the satellite control database. For example, you can grant this user ID the SELECT privilege on only the SATELLITES, TARGET_AUTH, and TARGETS tables. Consider doing this if, for example, you want to use a different authentication credential for the test synchronization than for a synchronization session in which the satellites execute group batches.

If the satellites are enabled to execute the group batches when they synchronize in test mode, they will download the group-level user ID and password that they require to authenticate with their satellite control server for synchronization.

Related reference:

- “GRANT (Database Authorities) statement” in the *SQL Reference, Volume 2*

- “GRANT (Table, View, or Nickname Privileges) statement” in the *SQL Reference, Volume 2*

Recommended Privileges to Grant on Stored Procedures and Bind Files

The following table lists the bind files and stored procedures, and the type of privileges that you should grant on them to your groups.

Bind File or Stored Procedure	Privilege for Administrative Group	Privilege for Group Authentication Credential	Privilege for Help-Desk Staff Group
DB2SATCS	REBIND/EXECUTE	EXECUTE	
DB2PROM	EXECUTE		

Creating a Group for the Satellites

A group consists of related satellites that share characteristics such as the database definition and the application that runs on the satellites. When you create the group, you must specify the authentication credential that you created in *Creating the User ID and Authentication Credential Required for Satellite Synchronization*. The satellites in the group will use this authentication credential when they synchronize.

Prerequisites:

The authentication credential that the group satellites will use to synchronize must already exist.

Procedure:

To create a group, use the Create Group window, which is available from the Satellite Administration Center. You specify the name of the authentication credential required for synchronization in the **Authentication name** field.

Related tasks:

- “Creating the User ID and Authentication Credential Required for Satellite Synchronization” on page 242
- “Creating a group : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Creating Test Satellites in the Satellite Administration Center

You create test satellites to perform the synchronization test. You also use them to test the group batches.

Prerequisites:

The group to which you want to assign the test satellites must already exist.

Procedure:

To create test satellites:

1. Use the Create Satellite notebook, which is available from the Satellite Administration Center, to create the satellites.

Notes:

- a. When you create a satellite, you must specify a unique identifier for it. This identifier must match the value that exists on the satellite. For more information, see [Setting the DB2SATELLITEID Registry Variable on a Satellite](#).
- b. You can use the subgroup attribute of satellites to further categorize the satellites. The subgroup can identify different regions, or any other grouping that you require. You specify the subgroup attribute in the **Subgroup** field of the Create Satellite notebook. You can also use the subgroup attribute to control the staging of a deployment. For example, you could enable one subgroup to synchronize before enabling the other subgroups.

When satellites are created, they are production satellites by default, and are not enabled to execute batches. Now change the satellite to a test satellite.

2. To change production satellites to test satellites:
 - a. In the Satellite Administration Center, open the satellite details view.
 - b. In the contents pane, select the satellites that you want to use as test satellites, and click the right mouse button.
 - c. Select **Set as Test Satellite** from the pop-up menu.

A window opens with the warning message SAT2022. Click **OK** to close the message window.

The satellites are now test satellites, but still cannot execute the group batches. You will enable these satellites to execute the group batches when you test the group batches.

Related tasks:

- “Setting the DB2SATELLITEID Registry Variable on a Satellite” on page 250

- “Creating a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Viewing satellite details : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Creating the Satellite Authentication File on a Satellite

The satellite authentication file, `satadmin.aut`, is created either when you install DB2 with the Satellite Synchronization component, or when the first test synchronization is run from the satellite.

Procedure:

To create the `satadmin.aut` file, issue the **db2sync -t** command on the satellite to perform a synchronization test. At this time, you will be prompted for the user ID and password required to authenticate with the satellite control server, and this authentication credential will be stored in the file.

Related tasks:

- “Creating the User ID and Authentication Credential Required for Satellite Synchronization” on page 242

Related reference:

- “db2sync - Start DB2 Synchronizer Command” in the *Command Reference*

Setting the Application Version on a Satellite

When a satellite synchronizes, it passes its application version to the satellite control server. The satellite control server uses this information, in conjunction with the group that the satellite belongs to, to determine which of the group’s application-version batches the satellite will execute. The application version allows you to support multiple versions of the application within a group, which enables you to perform a staged deployment of a new version of the application.

Prerequisites:

The Satellite Synchronization component must be installed on the DB2 server that you want to use as a satellite.

Procedure:

Use one of the following methods to set the application version on the satellite:

- Specify the application version of the satellite when you install DB2 with the Satellite Synchronization component.
- Specify the application version when you install the application on the satellite. If you specify the application version when you install the application, the install program must call either the `db2SetSyncSession` API or the `db2sync -s` command to record the application version.

Note: The application version is case sensitive.

You may find it more convenient to supply the application version when you install your initial test satellites. In a large production environment, you should specify the application version when you install the application.

Related concepts:

- “Application Versions and Batches in a Satellite Environment” on page xx

Related tasks:

- “Installing a New Version of the Application” on page 320
- “Setting the Application Version with the `db2SetSyncSession` API” on page 282

Related reference:

- “`db2sync - Start DB2 Synchronizer Command`” in the *Command Reference*
- “`db2SetSyncSession - Set Satellite Sync Session`” in the *Administrative API Reference*

Setting the DB2SATELLITEID Registry Variable on a Satellite

When a satellite synchronizes, it passes its ID to the satellite control server. The satellite ID is determined as follows:

1. If a value is specified for the `DB2SATELLITEID` registry variable, the satellite ID is this value.
2. Otherwise, the logon ID is used as the satellite ID.

The satellite control server verifies that the same value is recorded in the satellite control database. You create the satellite ID entry in the satellite control database when you create the satellite using the Create Satellite notebook in the Satellite Administration Center. The satellite ID must be recorded on both the satellite and in the satellite control database before synchronization can occur. Having the satellite ID on both the satellite and in the satellite control database ensures that only known satellites can synchronize.

If you do not want to use the logon ID as the satellite ID, you can set the value of the DB2SATELLITEID registry variable when you install DB2 with the Satellite Synchronization component. If you do not specify this registry variable during installation, you must specify it before synchronization can occur.

Procedure:

To set the DB2SATELLITEID registry variable, use the **db2set** command on the satellite as follows:

```
db2set DB2SATELLITEID='identifier' -i instance_name
```

Where:

identifier

Is the character string that uniquely identifies the satellite. The string can be a maximum of 20 characters (including blanks).

Notes:

1. Each satellite ID must be unique within a satellite control database.
2. The value that you specify on the satellite for *identifier* must match the satellite ID that you specified when you created the satellite with the Create Satellite notebook. To find the ID of a specific satellite, you can use the satellite details view in the Satellite Administration Center.
3. The satellite ID is case sensitive.

instance_name

Is the name of the instance on the satellite. Unless you have created another instance on the satellite, this value is db2.

The **db2set** command can be issued from the operating system command line, or from an application program; for example, an installation program for the end-user application.

Related tasks:

- “Creating a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Viewing satellite details : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Related reference:

- “db2set - DB2 Profile Registry Command” in the *Command Reference*

Verifying the Setup Before the Test Synchronization

When you are finished setting up the satellite control server and your test satellites, you should ensure that the configuration information in the satellite control database and on the satellite is both correct, and consistent.

Procedure:

Use the following table to verify that you have correctly set up the satellite control server and the satellite before having the satellite attempt a test synchronization.

Configuration Information	Satellite Control Database	Satellite
Satellite ID	Create the ID using the Create Satellite notebook. You can view the ID using the satellite details view.	The satellite ID can be either the user's logon ID, or the value of the DB2SATELLITEID registry variable on the satellite.
Authentication credential required for synchronization	Create the authentication credential using the Create Authentication window, and associate the credential with the group when you create the group.	The file satadmin.aut must exist on the satellite. Create the file using the db2sync -t command.
Application version	Create using Create Application Version window.	Set the application version on the satellite using either the db2sync -s command, or the db2SetSyncSession API. You can also specify the application version when you install DB2 with the Satellite Synchronization component on the satellite.

In addition, the versions of DB2 on both the satellite control server and the satellite must be compatible.

Related concepts:

- "Incompatible Versions of DB2 on the Satellite and the Satellite Control Server" on page 332

Testing the Synchronization Capability of a Satellite

After you have set up and configured the satellite control server and one or more test satellites, you can run a synchronization test to verify that the satellite can synchronize.

Procedure:

To test the synchronization capability of the satellite:

1. Open a command prompt window and enter the following command:

```
db2sync -t
```

The DB2 Synchronizer opens in test mode.

2. Click **Test**.

The Connect to Control Database window opens if you did not specify the user ID and password that the satellite requires to synchronize when you installed DB2.

3. If the Connect to Control database window opens, type the user ID and password that the satellite requires to synchronize, and click **OK**.

The Catalog Control Database window opens if the satellite control database is not yet cataloged at the satellite.

4. If the Catalog Control Database window opens, specify the required information about the instance where the satellite control database resides. This information includes the name of the instance, the host name of the machine where the instance resides, and the service name or port number of the instance. You can click the **Refresh** button to have DB2 discovery search for the systems that have DB2 installed on them.
5. When DB2 discovery completes, select the system that has the satellite control server from the drop-down list.
6. Click **Retrieve** to retrieve the list of DB2 instances on the system.
7. When the retrieve operation completes, select the satellite control server instance from the drop-down list.

When you select the satellite control server instance, the **Host name** and **Service name** fields are automatically updated with the values for that instance.

8. Click **OK**.

If the list of systems returned by DB2 discovery does not include the system on which the satellite control server resides, you must enter the host name for the system in the **Host name** field, and the port number in the **Service name** field. Then click **OK**.

If all the information specified at the satellite control server and at the satellite matches, and the satellite control database is correctly cataloged, the test synchronization will succeed. The next major task is to create the test batches that set up the database definition and data for the first version of the end-user application, then to test these batches.

Related tasks:

- “Creating and Testing Group Batches” on page 254

Related reference:

- “db2sync - Start DB2 Synchronizer Command” in the *Command Reference*

Creating and Testing Group Batches

Satellites execute group batches to create and maintain the database definition that is required by the application that runs on the satellite. When a test satellite can successfully perform a test synchronization (meaning that the test satellite can connect to the satellite control server and validate its configuration), you are ready to set up the test batches that create and maintain the database definition for the first version of the application. When these batches are set up, the test satellites will be able to execute them when they synchronize.

Prerequisites:

The test satellite must have successfully performed a test synchronization.

Procedure:

To create and test the group batches, perform the following steps:

1. Create authentication credentials.
2. Create execution targets.
3. Create an application version:
 - a. Create your first test level, level 0, for the application version.
 - b. Edit level 0 to add batches.
 - c. Add batch steps to the group batches.
4. Test the group batches

Related tasks:

- “Creating Authentication Credentials” on page 255
- “Creating Execution Targets” on page 256
- “Creating an Application Version for a Group” on page 257

- “Creating Level 0 of an Application Version” on page 258
- “Editing Level 0 of an Application Version to Create or Modify Group Batches” on page 259
- “Changing Batch Steps in a Group Batch” on page 260
- “Testing Group Batches” on page 262

Creating Authentication Credentials

You create authentication credentials for each DB2 instance and DB2 database that a satellite will either attach or connect to when it synchronizes. You can create a separate authentication credential for every potential target, or you can use one authentication credential for all targets, whichever is appropriate. For each batch step that the satellite executes when it synchronizes, the satellite will authenticate with the execution target using the user ID and password of the associated authentication credential.

For example, assume that you have a DB2 instance that is called DB2, and a DB2 database that is called DEPT. Assume that there is a user ID with SYSADM authority that is called DEPTADM with a password of DEPTPW. This is the user ID that you will associate with scripts that require SYSADM authority to execute. You could create the authentication credential associated with this user ID and password under the name DEPTCRED. This is the authentication credential that you would specify when you create an execution target.

Prerequisites:

DB2 uses the operating system security manager to authenticate clients. For this reason, the user ID and password must exist in the security manager of the operating system of the execution target. In the example above, the DEPTADM user ID must exist in the security manager with a password of DEPTPW. To enable this user ID to be able to execute the DB2 commands and SQL statements contained in the scripts, you must give this user ID SYSADM authority. Add this user to a group, then update the value of the database manager configuration parameter *sysadm_group* to be the group name.

Procedure:

To create authentication credentials, use the Create Authentication window. Enter the following values (the values follow the example above):

1. Specify DEPTCRED for the **Name** field.
2. Specify DEPTADM for the **User ID** field.
3. Specify DEPTPW for the **Password** and **Confirm password** fields.

4. Click **OK**.

After you create the authentication credentials, you can create the execution targets.

Related tasks:

- “Creating Execution Targets” on page 256
- “Creating an authentication : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Related reference:

- “System Administration Authority Group Name configuration parameter - sysadm_group” in the *Administration Guide: Performance*

Creating Execution Targets

All scripts execute against an execution target, which can be a DB2 instance, a DB2 database, or the operating system on the satellite. You must create a named execution target for every DB2 instance or DB2 database against which a script will execute. You do not need to create a named execution target for scripts that execute against an operating system.

When you have cataloged all of the instances and databases against which scripts will execute, you use the Create Target window in the Satellite Administration Center to create the named execution targets for these instances and databases.

Prerequisites:

Before you can create a named execution target for a DB2 instance or DB2 database, you must first catalog the instance or database in the Control Center.

Note: You must catalog the instance and the database using the same name and alias under which the instance and database is cataloged on the satellites.

Procedure:

Assume that you want to create execution targets for the DB2 instance and the DEPT database. On the Create Target window:

1. In the **Alias** field, select the target. You must select the same alias under which the instance is cataloged on the satellites. This is the same name under which the instance is displayed in the Control Center.

The **Type** field is automatically filled based on the type of the alias. In this situation, the type is Instance.

2. In the **Authentication name** field, select DEPTCRED. This is the name of the authentication credential that was created in Creating Authentication Credentials. This authentication name is associated with a user ID that has SYSADM authority.
3. Click **Test** and a test attachment is attempted to verify the user ID and password against the target for validation. Type the password for the user ID when you are prompted for it.

Note: The user ID and password must already exist at the target; otherwise, the test will fail.

Assuming that you want to use the same authentication credentials, follow the same procedure for the DEPT database. Again, remember to select DEPT in the **Alias** field for the target. The database alias must match the alias under which the database is cataloged at the satellites.

After you create the execution targets, you are ready to create the application version for the group.

Related concepts:

- “Requirement to Catalog Instances and Databases in the Control Center Instance” on page 229

Related tasks:

- “Creating Authentication Credentials” on page 255
- “Creating an Application Version for a Group” on page 257
- “Creating a target : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Creating an Application Version for a Group

An application version is associated with a set of setup, update, and cleanup batches that set up and maintain a specific database definition. The database definition applies to a specific version of the application that runs on the satellites of a group.

Prerequisites:

For a satellite to be able to execute the group batches of an application version, the satellite must have the same application version.

Procedure:

To perform this task, use the Create Application Version window, which is available from the Satellite Administration Center.

After you create the application version, the next step is to create the first level, level 0, in the application version.

Related concepts:

- “Application Versions and Batches in a Satellite Environment” on page xx
- “Life Cycle of an Application Version” on page 201

Related tasks:

- “Creating Level 0 of an Application Version” on page 258
- “Creating an application version : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Creating Level 0 of an Application Version

When an application version is first created, it contains no levels. The first level that you create in an application version is level 0. You initially use level 0 to contain the set of group batches that set up and maintain the database definition for the application. This level is created in the test state.

Prerequisites:

The application version must already exist.

Restrictions:

An application version can contain only one level 0.

Procedure:

To add level 0 to a new application version, use the Edit Application Version window, which is available from the Satellite Administration Center. The first time that you open this window, it is empty, as the application version does not yet have any levels associated with it:

When level 0 is first created, it is empty. No batches are associated with it. The next step is to edit level 0 of the application version create the group batches.

Related concepts:

- “Levels of an Application Version” on page 206
- “Life Cycle of an Application Version” on page 201

Related tasks:

- “Editing Level 0 of an Application Version to Create or Modify Group Batches” on page 259
- “Editing an application version : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Editing Level 0 of an Application Version to Create or Modify Group Batches

When level 0 is first created, it contains no group batches. When level 0 is in the test state, all the batches and batch steps you add are fully modifiable.

Procedure:

To perform this task:

1. Select level 0 in the Edit Application Version window, and click **Change**.
2. On each of the Setup, Update, and Cleanup pages of the Change Level notebook:
 - a. Specify the name of the batch.
 - b. Add or import the scripts that you want to be executed to the **Batch steps** field:
 - If you already have created scripts in the Satellite Administration Center, click **Add** to add them to the batch.
 - If you have saved scripts, click **Import** to import them from the Task Center to the batch.

You do not have to use all three types of batches. For example, in the initial phase of developing a database definition, you may not consider it necessary to have a cleanup batch. If a particular type of batch is not present when a satellite synchronizes, that batch type will be bypassed.

After you create the group batches for the group’s application version, you modify the batch steps of the group batches.

Related concepts:

- “Group Batches” on page 197
- “Levels of an Application Version” on page 206

Related tasks:

- “Changing Batch Steps in a Group Batch” on page 260
- “Editing an application version : Satellite Administration Center help” in the *Help: Satellite Administration Center*

- “Adding scripts to a batch : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Importing scripts to a batch : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Changing Batch Steps in a Group Batch

After you specify the scripts to be included in the batches that satellites execute when they synchronize, you must combine the scripts with the authentication credentials, execution targets, and success code sets that are required for the scripts to be executed. The combination of a script with the information that it requires for the satellite to be able to execute it is known as a batch step.

Procedure:

To perform this task, use the Change Batch Step notebook, which is available from the Satellite Administration Center.

Repeat the following steps for every script of every batch:

1. On the Edit Application Version window, select the new level, level 0, and click **Change**.
The Change Level notebook opens.
2. On the Change Level notebook, select the batch page for which you want change batch steps.
3. Select the batch step with which you want to associate an execution target, a success code set, and an authentication credential, and click **Change**.
The Change Batch Step notebook opens.
4. On the Script page:
 - a. Optional. Change the name of the script in the **Script** field.
 - b. Optional: Specify a description of the script in the **Description** field.
 - c. Optional. Change the type of target against which the script will execute from the **Type** radio buttons.
 - d. Optional. If you want to import additional scripts, click **Import**.
 - e. Optional. If you want, you can edit the contents of the script in the **Script contents** box.
 - f. Optional. If the script is parameterized, select the **Parameterized** check box.
 - g. Optional. To use a character other than the semicolon (;) as the statement termination character for a script that executes against a DB2 instance or a DB2 database, specify the character in the **Statement termination character** field.

5. For a script that executes against a DB2 instance or a DB2 database, on the Execution Target page:
 - a. Select the alias of the DB2 instance or DB2 database for the **Target alias** field. Use the ... button to display the list of targets.

Note: The alias that you select must be the alias that is cataloged at the satellite.
 - b. Select the named authentication credential for the **Authentication** field. The credentials are required by the satellite for its authentication with the target. Use the ... button to display the list of authentications.
6. On the Success Codes page:
 - a. Specify the name of the success code set in the **Success codes set name** field.

Note: If you have already created a success code set, you can click ... to list it.
 - b. Specify a description for the success code set in the **Description** field.
 - c. Specify the success code set for the script in the **Specify codes** box.
 - d. Click **Add** to add a success code relation pair to the **Specify codes** box.
7. Click **OK** to exit from the Change Batch Step notebook.
8. Click **OK** to exit from the Change Level notebook.

When you are finished changing the batch steps of the group batches, you test the group batches to check whether they produce the results that you want.

Related concepts:

- “Components of a Batch Step” on page 188
- “Parameterized Scripts” on page 193

Related tasks:

- “Testing Group Batches” on page 262
- “Changing the level of an application version : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Editing an application version : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Changing Batch Steps : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Importing scripts to a batch : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Testing Group Batches

When you have the group batches set up for the first level of the application version, you should test them on a small number of test satellites to ensure that the batches produce the results that you expect before you promote the batches to production.

Prerequisites:

All databases and instances that the satellite will access when executing the group batches must already be cataloged on the satellite.

Procedure:

To test group batches:

1. Enable the test satellites to execute batches. (These are the test satellites that were created in *Creating Test Satellites in the Satellite Administration Center*.) When satellites are first created, they are disabled, and cannot execute batches.

Note: It is recommended that you maintain the same set of test satellites throughout the life cycle of the application.

2. Synchronize the test satellites so they execute the test-level batches.
3. Check the results of the synchronization session.

Related tasks:

- “Enabling Test Satellites to Execute Test-Level Batches” on page 262
- “Creating Test Satellites in the Satellite Administration Center” on page 248
- “Synchronizing Test Satellites to Execute Test-Level Batches” on page 263
- “Checking the Results of the Synchronization Session” on page 264
- “Cataloging Instances and Databases on Test Satellites” on page 236

Enabling Test Satellites to Execute Test-Level Batches

When satellites are created, they are, by default, unable to execute batches. Before the satellites can synchronize, you must enable them.

Procedure:

To enable test satellites to execute test-level batches:

1. In the Satellite Administration Center, open the satellite details view for the group.

2. In the contents pane, select the test satellites and click the right mouse button.
3. Select **Enable** from the pop-up menu. Click **OK** when you are prompted for confirmation.

The test satellites are now enabled to execute test batches.

After the test satellites are enabled to execute batches, the next step is to have the test satellites synchronize so that they download and execute the test-level group batches.

Related tasks:

- “Synchronizing Test Satellites to Execute Test-Level Batches” on page 263
- “Viewing satellite details : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Synchronizing Test Satellites to Execute Test-Level Batches

After the test satellites are enabled to execute batches, they can synchronize to download and execute the test batches.

Procedure:

To synchronize the satellite:

1. Open the DB2 Synchronizer application using one of the following methods:
 - Click **Start** and select **Programs->IBM DB2->Set-up Tools->Satellite Synchronizer**.
 - Open a command prompt window and enter the following command:
`db2sync`

The DB2 Synchronizer application opens.

2. Click **Start**.

The status indicator bar shows the progress of the synchronization.

If a problem occurs, you can click **Log** to open the Log Information window. This window shows the detailed log messages that are written during the synchronization session.

3. When the synchronization session is complete, click **Close**.

Note: The first time that a satellite synchronizes to execute its group batches, its application version is recorded in the satellite control database. That is, the satellite details view in the Satellite Administration Center does

not display the satellite's application version until it synchronizes. You cannot use the Satellite Administration Center to specify the application version of the satellite.

After the synchronization session ends, check the results of the synchronization session to verify that the group batches result in the correct database definition on the test satellites.

Related tasks:

- "Checking the Results of the Synchronization Session" on page 264

Checking the Results of the Synchronization Session

After the satellite synchronizes, you should check the results of the synchronization session to ensure that no unexpected errors or warnings were issued that were not accounted for by the success code sets of the batch steps.

Procedure:

To determine the results of the synchronization session, perform the following steps:

1. Open the satellite details view, and locate the test satellites that have executed the batches. In particular, check the values under the **State** column. If the entry in this column is **Results stored**, the satellite has executed the batches and reported back the results of the execution. If the entry in this column is **Satellite failure**, an error occurred when the satellite was synchronizing. The satellite does not execute any batch steps after encountering the error.
2. Regardless of whether or not an error occurs, you should check the logs of the test satellites that synchronized. To check the logs:
 - a. From the satellite details view, select a test satellite and click the right mouse button.
 - b. Select **View details** from the pop-up menu. The Log Details window opens for the satellite. If the satellite reported a failure, the last log entry for the satellite indicates the nature of the failure, and supplies additional information about the failure for diagnosis.

If a failure occurs, it is easy to identify which batch step you will have to modify, if necessary. If, however, no satellites report a failure, you should still examine the database definition on the test satellites to determine whether the batches produced the results that you intended. If any problems occurred, you need to fix the problems caused by the test-level group batches. If the batches

produced the results that you intended, and no satellites reported an error when executing the group batches, you can promote the test-level batches to production.

Related tasks:

- “Promoting the Batches of Test Level 0 to Production” on page 267
- “Fixing Problems Caused by Test-Level Group Batches” on page 265
- “Viewing log details : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Viewing satellite details : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Fixing Problems Caused by Test-Level Group Batches

When you examine the results of the synchronization session on the test satellites, you may find that a satellite reported an error when executing the test-level group batches, or that the database definition created by the group batches does not adequately support the requirements of the application.

Procedure:

If a failure occurred when the test satellite executed the group batches, or the results of the synchronization were not satisfactory, you can use one or more of the following actions to correct the situation using the Satellite Administration Center:

- Correct the test batch step that produced the incorrect results. The next time the satellite synchronizes, it will begin executing the associated batch from this batch step. This method is appropriate if the batch step that failed did not leave the satellite in an inconsistent state, or did not produce an inappropriate change to the database definition. As an example, assume that you have an SQL statement that fails because of incorrect syntax. This error will not result in changes to the database definition on the satellite that must be undone. If, however, the SQL statement did run and resulted in a table being populated with incorrect data, this method would not be appropriate. Instead, you would use one of the following methods.

Note: If the test satellite failed, you must promote it before it can execute its group batches.

- In some situations, you must undo the results of the test batch step or steps. In this situation, you can use a fix batch to undo the results of specific batch steps:
 1. Create a fix batch to undo the results of one or more batch steps.
 2. Assign the fix batch to the test satellite and synchronize again.

3. When the satellite has successfully executed the fix batch (and the batch steps that caused the problem have been corrected), promote the satellite back to executing its group batches. When you promote the satellite, set the batch step at which the test satellite will resume executing the batch to be the first step of the batch that failed. In this way, the test satellite will re-execute the scripts when it synchronizes
- You can also use a fix batch to undo not only the effects of the test batch steps in error, but also of all the batch steps that the test satellite has already executed. In this situation, after you fix the test batch steps, set the execution starting point to step 1, and the test satellites will execute all the test batch steps in sequence the next time they synchronize.

After fixing all the problems that occurred, you are ready to have the test satellites execute the test batches again. The next time that the satellites synchronize, they will begin executing each batch at the batch step that you specified when you set the execution starting point, or the batch step they stopped at when they last synchronized.

After the test satellites synchronize, check the satellite details view and the logs of the test satellites again. If an error occurred, or individual batch steps did not produce the required results, perform the necessary corrective action, and repeat the synchronization. Repeat this process until you are satisfied with the results that you obtain. When the database definition that is produced by the batches meet your requirements, you are ready to promote level 0 and its batches to production.

Related concepts:

- “Fix Batches” on page 219

Related tasks:

- “Assigning a Fix Batch to the Satellite” on page 340
- “Promoting the Batches of Test Level 0 to Production” on page 267
- “Setting the Execution Starting Point for a Satellite” on page 267
- “Identifying and Fixing a Failed Satellite” on page 337
- “Promoting a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Setting the execution starting point for a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Viewing satellite details : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Promoting the Batches of Test Level 0 to Production

When you are satisfied with the results of the test-level group batches, promote level 0 and its batches from test to production. Promoting level 0 makes the group batches of level 0 available to the production satellites, that is, when they are created and enabled to execute batches. When the production satellites synchronize, they can download and execute these batches to set up the production environment. (The test satellites will no longer be able to execute the batches of level 0.)

Procedure:

To promote the batches in a test level to production, use the Edit Application Version window, which is available from the Satellite Administration Center.

Related tasks:

- “Editing an application version : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Setting the Execution Starting Point for a Satellite

When you set the execution starting point for a satellite, you specify from which batch step the satellite executes batches when the satellite synchronizes.

Procedure:

You set the execution starting point for a satellite for one of two reasons:

- You are using test satellites to execute the test-level group batches. In this situation, you may need to reset the execution starting point. To perform this task, use the Edit Satellite notebook, which is available from the Satellite Administration Center.
- You are migrating a satellite that was already running DB2, but not a member of a group.

If you have satellites that were already configured when they joined the group, you will probably want them to begin executing the group-level batches of the application version at different batch steps than those satellites that have not yet executed batches, or are not yet properly configured.

To perform this task, use the Set Execution Starting Point window, which is available from the Satellite Administration Center.

Note: If you are setting the execution starting point for multiple satellites (that is, you selected multiple satellites from the satellite details view in the Satellite Administration Center, then selected **Set Execution Starting**

Point from the pop-up menu), the behavior of the Set Execution Starting Point window depends on whether the selected satellites have the same application version:

- If all the satellites have the same application version, the batch fields are pre-filled with the name of the batches associated with the application version. The batch step field for each batch is set to 1. This field cannot be edited.

The ... button for each batch that has more than one batch step is enabled, however, so that you can specify a different batch step for the satellites to begin executing the batch.

- If all the satellites do not have the same application version, each batch field displays Not Applicable, and each batch step field is set to 1.

The ... button for each batch is disabled. In this situation, however, the batch step fields are editable.

Related tasks:

- “Editing a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Setting the execution starting point for a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Chapter 11. Working with the Model Office

The Model Office and the Development and Acceptance-Testing Phase	269	The Model Office and the Production-Deployment and Post-Deployment Phases	274
Role of the Model Office in the Development and Acceptance Testing Phase	269	Model Office During the Production-Deployment Phase	274
Characteristics of a Model Office.	270	Model Office in the Post-Deployment Phase	275
Installing and Setting up the Model Office	271	Uses of the Model Office During the Post-Deployment Phase.	275
Synchronizing the Model Office to Test the Group Batches	273		

A model office is a special member of the test satellites of a group, as it is representative of the satellites in its group. You typically have one model office for each version of the application that is used in the group. Because the model office is representative of your satellites, you can use it perform administrative tasks for the group. For example, you can use the model office to recreate a problem that occurs on a production satellite. When you recreate the problem on the model office, you can then use it to create and test a fix for the problem.

The administrative tasks that you can perform using the model office differ depending on whether you are in the development and acceptance-testing phase, the production-deployment phase, or the post-deployment phase of an application version for a group. The sections that follow describe the different ways in which you can use the model office in the different phases.

The Model Office and the Development and Acceptance-Testing Phase

The topics that follow describe how to use the model office in the development and acceptance testing phase of the group batches of the application version.

Role of the Model Office in the Development and Acceptance Testing Phase

When you build the model office, it becomes the model of the test and production satellites that you later deploy for a specific group. Although the model office does not have to be installed on the same hardware that you intend to use with the other satellites of the group, the model office should match its group satellites on several characteristics:

- The same installed components.

When you install DB2[®] on the satellite that will be the model office, you should install the same components that you intend to install on the other

satellites of the group. This shared characteristic is important when you use the model office as the model for deploying production satellites.

- The same DB2 database definition.

The model office should have a DB2 instance with the same database manager configuration values, and a DB2 database with the same name and database configuration values that you intend to use on your production satellites. The model office should also have the same connection information (node, database and DCS directory information) as the other satellites. The connection information should include the definitions that are required to connect to the satellite control database on the satellite control server.

- The version of the application. You should install the same version of the application on the model office that you intend to install on the group satellites. You set the application version on the model office to ensure that, when it synchronizes, it executes the scripts for the correct version of the application. Because the model office is a model of the deployed satellites, you should test the model office to ensure that it behaves similar to the other satellites.

Related concepts:

- “Model Office During the Production-Deployment Phase” on page 274
- “Model Office in the Post-Deployment Phase” on page 275

Characteristics of a Model Office

You build the model office to represent the initial version of an application during the development and acceptance-testing phase. The model office should match the group satellites on several characteristics:

- It should have the same DB2[®] components installed on it.
- It should have the same DB2 database definition.
- It should have the specific version of the application installed on it that you are deploying to the production satellites.

The model office should be at this state before you begin the production-deployment phase.

Related concepts:

- “Role of the Model Office in the Development and Acceptance Testing Phase” on page 269
- “Model Office During the Production-Deployment Phase” on page 274
- “Model Office in the Post-Deployment Phase” on page 275

Related tasks:

- “Installing and Setting up the Model Office” on page 271

Installing and Setting up the Model Office

When you install and set up the model office for a group, it must resemble the production satellites that you intend to deploy in the group.

Procedure:

Perform the following steps to install and set up the model office:

1. Install the Windows-based operating system that you want to use on the model office.
2. Install DB2 on the model office. You should install the same components that you intend to install on the production satellites:
 - On Windows-based platforms, DB2 Personal Edition, DB2 Workgroup Server Edition, and DB2 Enterprise Server Edition can all function as satellites.
 - So that a satellite can synchronize, you must install the Satellite Synchronization component when you install DB2.
 - When you install DB2, you can:
 - Perform an interactive installation:
 - DB2 Workgroup Server Edition or DB2 Enterprise Server Edition
 - DB2 Personal Edition
 - Perform a response file installation
If you install the model office using a customized response file, you can provide values for:
 - The satellite ID
 - The user ID and password that are required to connect to the satellite control server and the satellite control database during synchronization
 - The user ID and password required for the Remote Command Service on the Windows NT and Windows 2000 platforms
 - You can also set the application version when you install DB2. You may find it easier, however, to set the application version when installing the application.
3. Catalog the remote databases and instances, including the satellite control server and the satellite control database, on the model office. See *Cataloging Remote Instances and Databases on the Model Office* and *Cataloging Remote Instances and Databases on the Model Office Using a Client Profile (Windows)* for information that you can use to simplify setting up the connection information.
4. Create the database

5. Install the application. The installation application should set the application-version information that is required for synchronization. If the installation of the application does not set the application version, set it by using the **db2sync -s** command.
6. If the installation of application does not create the required tables, indexes and other database objects that are required by the application, create them.
7. Perform any specific DB2 customization that is required to run the application. For example:
 - You can tune the environment for the application by setting database manager configuration parameters, database configuration parameters, and DB2 registry variables.
 - Set any CLI/ODBC values that are required for the application. Consult the application documentation or the application developer for specific requirements.

After you install and set up the model office, you can synchronize it to test the group batches.

Related concepts:

- “Configuration parameter tuning” in the *Administration Guide: Performance*

Related tasks:

- “Starting the DB2 Setup wizard for a DB2 server installation (Windows)” on page 23
- “Starting the DB2 Setup wizard (Windows)” on page 105
- “Response file installation of DB2 on Windows” on page 129
- “Customizing the Generated Response File for a Mass Installation” on page 313
- “Cataloging Remote Instances and Databases on the Model Office” on page 232
- “Cataloging Remote Instances and Databases on the Model Office Using a Customized Client Profile (Windows)” on page 233
- “Synchronizing the Model Office to Test the Group Batches” on page 273

Related reference:

- “UPDATE DATABASE CONFIGURATION Command” in the *Command Reference*
- “UPDATE DATABASE MANAGER CONFIGURATION Command” in the *Command Reference*
- “db2set - DB2 Profile Registry Command” in the *Command Reference*
- “UPDATE CLI CONFIGURATION Command” in the *Command Reference*

- “db2sync - Start DB2 Synchronizer Command” in the *Command Reference*
- “Configuration parameters summary” in the *Administration Guide: Performance*
- “General registry variables” in the *Administration Guide: Performance*

Synchronizing the Model Office to Test the Group Batches

When you finish installing and configuring the model office, you can synchronize it to test the group batches.

Prerequisites:

The satellite control server must already be installed and set up to support synchronization as described in *Preparing for a Synchronization Test*. In addition, the application version must exist and it must contain test-level group batches. The model office will execute these batches when it executes.

Procedure:

To synchronize the model office:

1. Create the model office in the satellite control server by using the Create Satellite window in the Satellite Administration Center. You should set up the model office as a test satellite.

You created a group when preparing for a synchronization test. The group that you created was the group intended for use for your production deployment. When you use the Create Satellite window to add the model office to the group, provide a name for the model office that conforms to the conventions that you intend to use for your model offices. You can use the **Description** field, and possibly the **Subgroup** field, to provide additional identification about the model office. When you complete your acceptance testing, you will use this model office to support the production-deployment and post-deployment phases for the application version in the group.

2. Run a synchronization test from the model office.

Run the test synchronization to verify that the model office can attach to the satellite control server and connect to the satellite control database.

3. Enable the model office to execute group batches.
4. Synchronize the model office to execute the test-level group batches.
5. Check the results of the synchronization session.

You must completely test the database definition that is created on the model office by the group batches. You may have to modify the group batches more than once before they produce the correct results.

After you have used the model office to complete the development and acceptance testing phase for you application version, the setup of the model office should match what you intend to deploy on your production satellites.

Related concepts:

- “Model Office During the Production-Deployment Phase” on page 274
- “Model Office in the Post-Deployment Phase” on page 275

Related tasks:

- “Preparing for a Test Synchronization” on page 240
- “Creating Test Satellites in the Satellite Administration Center” on page 248
- “Testing the Synchronization Capability of a Satellite” on page 253
- “Testing Group Batches” on page 262
- “Synchronizing Test Satellites to Execute Test-Level Batches” on page 263
- “Checking the Results of the Synchronization Session” on page 264

The Model Office and the Production-Deployment and Post-Deployment Phases

The sections that follow describe how to use the model office in the production-deployment phase and later.

Model Office During the Production-Deployment Phase

You can use the model office to generate the files that are required for automating the deployment of the group satellites. The installation process uses the generated files to install and set up the satellites. To generate the files, use two utilities that are provided by DB2:

- Use the response file generator to *reverse engineer* the installed model office, which produces an installation response file. You use the response file to install exact duplicates of the model office. When you perform the installation, DB2® is installed with the same components and configuration values that are on the model office.

Note: The term *reverse engineer* refers to the process of extracting information from an existing system and producing a script from that system. The script can then be used to recreate a similar or exact copy of the original system.

- Use the client profile export utility to generate a configuration file. This file contains configuration and connectivity information about the model office. When you import the configuration file to the satellite, the configuration and connection data for the model office is recreated on the satellite.

When using the DB2 Universal Database distributed installation process, you can specify a client profile to import, which facilitates complete automation of

the installation and configuration process. In this way, you can deploy all of your production satellites with a minimum of difficulty.

Related reference:

- “db2cfimp - Connectivity Configuration Import Tool Command” in the *Command Reference*
- “db2cfexp - Connectivity Configuration Export Tool Command” in the *Command Reference*
- “db2rspgn - Response file generator” on page 124

Model Office in the Post-Deployment Phase

After the group’s satellites are deployed, the model office reflects the current state of a typical group satellite, though some differences may exist between the model office and a satellite. For example, the hardware configuration or the operating system may be different on the model office.

Because the model office is configured to run one version of the application that is being used by the group satellites, you can use the model office to recreate the problems that occur on a production satellite. This means that you can use the model office to define and test fix batches to correct these problems.

The model office is vital, as it provides existing tools, like the Control Center, a working and representative database against which to operate. You can generate scripts using the wizards, notebooks, and windows of the Control Center against the model office. You then submit the scripts for execution against the model office and examine the changes caused by them, without affecting any production satellites. When you are satisfied with the results that the scripts produce, you can add the scripts to the group batches (or to a fix batch), as required. The scripts will be executed by the group satellites when they synchronize.

You should take backup images of the model office. If a change you make while testing a fix batch for a group satellite produces unexpected results, you can use the backup image to restore the model office. After the model office is restored, it will be at a known state, and you can try a different fix.

Related concepts:

- “Recovery of the Model Office and Test Satellites” on page 298

Uses of the Model Office During the Post-Deployment Phase

When the production satellites are deployed, unanticipated problems can occur. For example, a performance problem could occur. Because the model

office represents the deployed satellites, you can use it to investigate the problem, and to develop a test script to correct it.

- Using tools such as the Control Center and the command line processor (CLP), you can examine the model office and determine which changes are required to address a production problem.
- You can use the Control Center to make changes to the model office to correct problems. Instead of using the Control Center tools to complete the changes, use the Show SQL and Show Command options to generate and save scripts. You can then create a new test level of the batches for the application version, and add the scripts to the setup, update and cleanup batches, as required. The next time the model office synchronizes, it will execute the new batch steps. You can determine the effect of the changes by reviewing the execution logs, and by using the Control Center tools to examine the model office. If the changes are correct, the next step would be to promote the test level to production for the production satellites to execute.

The model office has already executed the batches that you promoted to production. When the rest of the group satellites next synchronize and execute the new batch steps, the model office will represent the group satellites that are at the same application version. If another problem occurs, you can again use the model office to investigate the problem, and to develop and test scripts to correct the problem.

Chapter 12. Developing a Synchronizing Application

How a Synchronizing Application Works	277	Managing Synchronization Sessions Using APIs	285
Setting up the Environment and the Development Machine	278	Synchronizing a Satellite with DB2 APIs	285
Setting up the Environment for a Synchronizing Application	278	Initiating a Synchronization Session Using the db2SyncSatellite API	287
Cataloging the Satellite Control Database on a Development Machine	279	Querying Synchronization Session Progress Using the db2QuerySatelliteProgress API	287
Binding Utilities on the Satellite to the Satellite Control Database	280	Stopping a Synchronization Session Using the db2SyncSatelliteStop API	288
Programming the Application	281	Building and Running Synchronizing Applications	289
Programming a Synchronizing Application	281	Building and Running Your Synchronizing Application	289
Setting the Application Version with the db2SetSyncSession API	282	Using the DB2 Synchronizer Application	290
Retrieving the Application Version with the db2GetSyncSession API	283		
Testing the Ability of the Satellite to Synchronize Using the db2SyncSatelliteTest API	284		

The sections that follow describe how to develop a synchronizing application for use in a satellite environment.

How a Synchronizing Application Works

A satellite requires a synchronizing application for the satellite environment. This could be separate from the satellite's user application, but normally the user application run by the satellite is also a synchronizing application.

The following figure shows how a synchronizing application that you build can interface with the database on the satellite and the satellite control database on the satellite control server in a development environment.

Typically, the user application will access the satellite's own database manager instance to manipulate database table data. This user application will normally also include the functionality of a synchronizing application, by using DB2's satellite APIs. The synchronization work of the user application includes accessing the satellite control server database manager instance, and through it, the satellite control database tables.

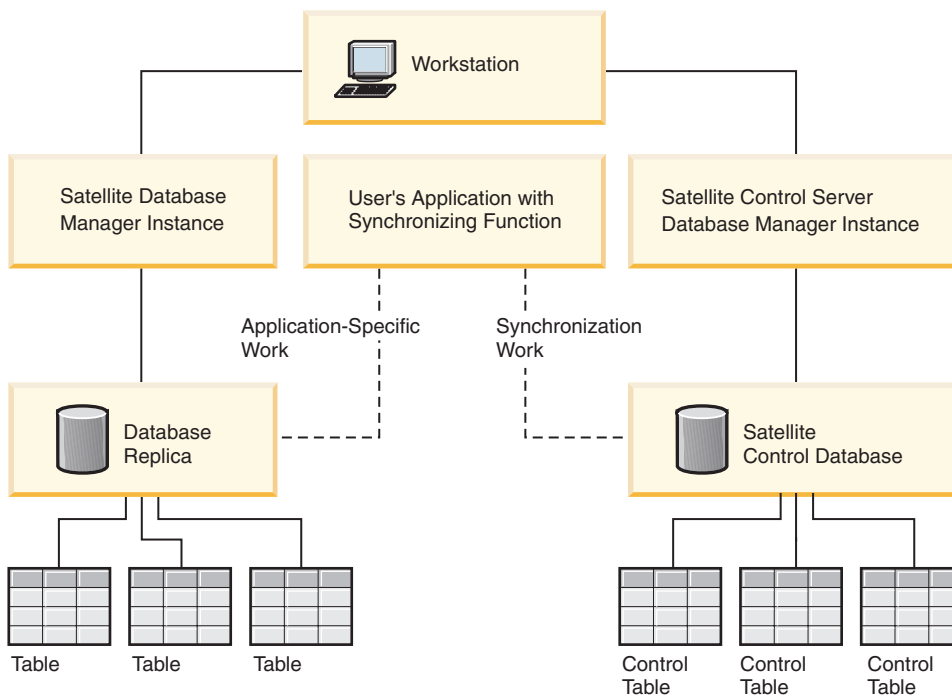


Figure 4. Synchronizing Application Program

Related tasks:

- “Setting up the Environment for a Synchronizing Application” on page 278
- “Building and Running Your Synchronizing Application” on page 289
- “Programming a Synchronizing Application” on page 281

Setting up the Environment and the Development Machine

The sections that follow describe the tasks that you need to perform before developing a synchronizing application.

Setting up the Environment for a Synchronizing Application

To build a synchronizing application for the satellite environment, you can use either the DB2 Personal Developer’s Edition, or the DB2 Universal Developer’s Edition. Either product includes the tools that you need to build applications that run on Windows operating systems.

Prerequisites:

The development environment for a synchronizing application requires the following:

- A DB2 supported Windows operating system.
- A DB2 supported compiler. A supported version of the Microsoft Visual C++ compiler is required for compiling DB2 satellite APIs, which are only available in C.
- DB2 Version 8 installed, including the Satellite Synchronization component and the AD Client.
- A client connection to the satellite control server using TCP/IP.
- A local satellite database.
- A satellite control server.
- The satellite control database (SATCTLDB) on the satellite control server.

Procedure:

To set up your development environment, do the following:

1. Catalog the satellite control database
2. Bind the utilities to the satellite control database

Related concepts:

- “DB2 Developer’s Edition Products” in the *Application Development Guide: Programming Client Applications*

Related tasks:

- “Cataloging the Satellite Control Database on a Development Machine” on page 279
- “Binding Utilities on the Satellite to the Satellite Control Database” on page 280

Related reference:

- “Windows Supported Software for Building and Running Applications” in the *Application Development Guide: Building and Running Applications*

Cataloging the Satellite Control Database on a Development Machine

You catalog the satellite control database on the satellite. You do not need to catalog the database on the satellite control server as it is automatically cataloged on the server when the database is created.

When you catalog the satellite control database, the database directory on the satellite is updated with the alias of the database that the application needs to access using a CONNECT statement. When the database manager processes requests from the application on the satellite, it uses the database alias to both find and connect to the database.

To access the satellite control database on the satellite control server, both the satellite control server and the satellite control database must be cataloged on the satellite. In addition, the satellite must have the correct authentication credentials to access the satellite control database.

Procedure:

To set up the catalog and authentication information, do one of the following:

- Enter the **db2sync -t** command to open the DB2 Synchronizer Application in test mode
- Manually catalog the satellite control database and the node on which it resides by issuing the CATALOG NODE and CATALOG DATABASE commands on the satellite:

```
db2 catalog tcpip node nodename remote hostname server service-name  
db2 catalog database satctlldb at node nodename
```

Related tasks:

- “Using the DB2 Synchronizer Application” on page 290

Related reference:

- “CATALOG DATABASE Command” in the *Command Reference*
- “CATALOG TCP/IP NODE Command” in the *Command Reference*
- “db2sync - Start DB2 Synchronizer Command” in the *Command Reference*

Binding Utilities on the Satellite to the Satellite Control Database

If the satellite is remote to the satellite control server, or is running a different version of DB2, or runs a different operating system, you need to bind the database utilities on the satellite, including the DB2 CLI, to the SATCTLDB database on the satellite control server.

Binding creates the package that the database manager needs to access the database when an application is executed. Binding can be done explicitly by specifying the BIND command against the bind file that is created during precompilation.

Procedure:

To bind the database utilities, do the following on the satellite:

1. Click **Start** and select, for example, **Programs -> IBM DB2 -> Command Line Tools -> Command Window**

The command window opens.

2. Connect to the SATCTLDB database. At the prompt, enter:

```
db2 connect to satct1db
```

3. Bind the utilities to the database by entering the following BIND command:

```
db2 bind %DB2PATH%\bnd\@db2ubind.lst blocking all  
sqlerror continue messages bind.msg grant public
```

Where %DB2PATH% is the path where DB2 is installed.

4. Exit the command window, and verify that the bind was successful by checking the bind message file, bind.msg. This file is in the path specified by %DB2PATH%.

Related reference:

- “BIND Command” in the *Command Reference*

Programming the Application

The sections that follow describe the tasks that you need to perform when developing a synchronizing application.

Programming a Synchronizing Application

DB2 provides a set of satellite APIs to use for programming your synchronizing application. You also need to set the satellite ID as this value is required for running the APIs.

The sample program, db2sat.c, in the sql1lib\samples\c directory demonstrates several of the Satellite APIs.

Procedure:

To program your synchronizing application, you need to set the unique satellite ID by using the **db2set** command to set the value of the DB2SATELLITEID registry variable, or you use the logon ID of the satellite user as the satellite ID. Follow the instructions in:

- Setting the DB2SATELLITEID Registry Variable on a Satellite

The following explain how to program the satellite APIs:

1. Setting the Application Version with the db2SetSyncSession API
2. Testing the Ability of a Satellite to Synchronize with the db2SyncSatelliteTest API
3. Synchronizing a Satellite with DB2 APIs

Related tasks:

- “Setting the DB2SATELLITEID Registry Variable on a Satellite” on page 250

- “Setting the Application Version with the db2SetSyncSession API” on page 282
- “Testing the Ability of the Satellite to Synchronize Using the db2SyncSatelliteTest API” on page 284
- “Synchronizing a Satellite with DB2 APIs” on page 285

Setting the Application Version with the db2SetSyncSession API

Before a satellite can synchronize with its satellite control server, the application version must be configured in the satellite’s environment.

When setting the application version, the most common return codes are:

- SQL3942I Synchronization session identifier was set successfully for the satellite.
- SQL3943N Synchronization session identifier exceeds the maximum length of 18 characters
- SQL3944I Synchronization session identifier was reset successfully for the satellite.
- SQL3946N Synchronization session identifier operation failed.

Procedure:

To set the application version:

- Use the db2SetSyncSession API to set the value of the application version on the satellite:

```
SQL_API_RC SQL_API_FN
db2SetSyncSession (      /* Set synchronization session */
    db2UInt32 versionNumber, /* Database version number */
    void * pParmStruct,    /* Input parameters */
    struct sqlca * pSqlca); /* SQLCA */
```

Note: The application version is case sensitive.

Define the application version size to its maximum length of 18:

```
#define SQL_SYNCSESSIONID_SZ 18
```

- Use the following structure for pParmStruct to define the string:

```
typedef struct db2SetSyncSessionStruct
{
    char          *piSyncSessionID; /* ID for sync session */
} db2SetSyncSessionStruct;
```

If you supply an empty string for piSyncSessionID, you will reset the satellite’s application version to NULL.

Related tasks:

- “Retrieving the Application Version with the db2GetSyncSession API” on page 283

Related reference:

- “db2SetSyncSession - Set Satellite Sync Session” in the *Administrative API Reference*

Retrieving the Application Version with the db2GetSyncSession API

When retrieving the application version, the most common return codes are:

- SQL3945I Synchronization session identifier for the satellite was retrieved successfully.
- SQL3946N Synchronization session identifier operation failed.

Procedure:

To retrieve the application version:

- Use the db2GetSyncSession API to query the value of the application version on the satellite:

```
SQL_API_RC SQL_API_FN
db2GetSyncSession (          /* Get sync session ID   */
    db2UInt32 versionNumber, /* Database version number */
    void * pParmStruct,      /* Output parameters     */
    struct sqlca * pSqlca);  /* SQLCA                 */
```

- Use the following structure for pParmStruct to retrieve the value:

```
typedef struct db2GetSyncSessionStruct
{
    char          *poSyncSessionID; /* ID for sync session */
} db2GetSyncSessionStruct;
```

- Supply a buffer at least as large as SQL_SYNCSESSIONID_SZ (+1 for the NULL terminator) to receive the application version.

Related tasks:

- “Setting the Application Version with the db2SetSyncSession API” on page 282

Related reference:

- “db2GetSyncSession - Get Satellite Sync Session” in the *Administrative API Reference*

Testing the Ability of the Satellite to Synchronize Using the db2SyncSatelliteTest API

When testing the ability of a satellite to synchronize, the most common return codes are:

- SQL3911I Test synchronization session completed successfully.
- SQL3931W The test synchronization session completed successfully. The satellite ID, however, could not be found in the satellite control database.
- SQL3932W The test synchronization session completed successfully. The satellite application version, however, is not set locally, or does not exist for this satellite's group at the satellite control server.
- SQL3933W The test synchronization session completed successfully. The release level of the satellite, however, is not supported by the release level of the satellite control server.
- SQL3934W The test synchronization session completed successfully. This satellite, however, is disabled at the satellite control server.
- SQL3935W The test synchronization session completed successfully. This satellite, however, is in the failed state at the satellite control server.
- SQL3955N The satellite control database name or its alias could not be found.

Procedure:

To test the ability of a satellite to synchronize with its satellite control server, execute the db2SyncSatelliteTest API on the satellite:

```
SQL_API_RC SQL_API_FN          /* Test the satellite's ability */
db2SyncSatelliteTest (         /* to synchronize                */
    db2UInt32 versionNumber,    /* Database version number       */
    void * pParmStruct,        /* Input/Output parameters      */
    struct sqlca * pSqlca);    /* SQLCA                          */
```

Because the satellite ID and application version are retrieved from the satellite, no parameter structure (pParmStruct) is required.

This API validates the following required configuration elements, which are necessary for synchronization to occur:

- The satellite control database, SATCTLDB, is cataloged at the satellite.
- The release level of DB2 on both the satellite and its satellite control server are compatible.
- The satellite ID is set on the satellite.
- The application version is set on the satellite.
- The satellite can communicate with the satellite control server.

- The authentication credentials that the satellite uses for authentication with the satellite control server are correct.
- The satellite is defined at the satellite control server.
- The satellite is enabled for batch execution at the satellite control server.
- The satellite is not in the failed state at the satellite control server.
- The satellite's application version exists for the satellite's group at the satellite control server

Related reference:

- “db2SyncSatelliteTest - Test Satellite Sync” in the *Administrative API Reference*

Managing Synchronization Sessions Using APIs

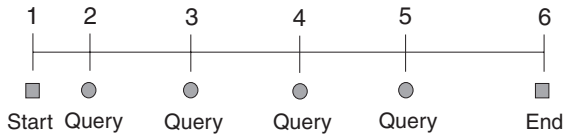
The sections that follow describe how to manage a synchronization session using the satellite APIs.

Synchronizing a Satellite with DB2 APIs

The following figure shows the events that can occur when a satellite is synchronized, including the APIs that can be called from the synchronizing application. Specifically:

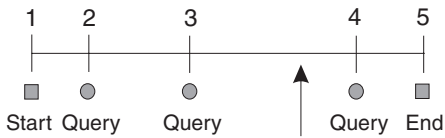
1. The db2SyncSatellite API is invoked, which starts the synchronization session. This API does not return until the session completes.
2. The db2QuerySatelliteProgress API is invoked, which returns information about the progress of the synchronization session. This API can be invoked more than once to return progress information. Invoke this API on another thread or process.
3. The db2SyncSatellite API returns when the synchronization session ends. The API can return normally, or the db2SyncSatelliteStop API can be invoked (on another thread or process) during a synchronization session to interrupt the session. When invoked, the db2SyncSatelliteStop API issues a STOP request to the db2SyncSatellite API. The db2SyncSatellite API, however, does not return until it finishes executing the current step of the synchronization process.

Uninterrupted Synchronization Session



Query is issued from a separate thread or process.

Interrupted Synchronization Session



Stop issued from a separate thread or process.

- db2SyncSatellite
- db2QuerySatelliteProgress
- ⬮ db2SyncSatelliteStop

Figure 5. Synchronization Time Line

Procedure:

To understand how to program these APIs, follow the instructions in these topics:

- Initiating a Synchronization Session
- Querying the Progress of a Synchronization Session
- Stopping a Synchronization Session

Related tasks:

- “Initiating a Synchronization Session Using the db2SyncSatellite API” on page 287
- “Querying Synchronization Session Progress Using the db2QuerySatelliteProgress API” on page 287
- “Stopping a Synchronization Session Using the db2SyncSatelliteStop API” on page 288

Related reference:

- “db2QuerySatelliteProgress - Query Satellite Sync” in the *Administrative API Reference*

- “db2SyncSatellite - Sync Satellite” in the *Administrative API Reference*
- “db2SyncSatelliteStop - Stop Satellite Sync” in the *Administrative API Reference*

Initiating a Synchronization Session Using the db2SyncSatellite API

To initiate a synchronization session, you must invoke the db2SyncSatellite API from the context of the satellite’s instance (the same as a synchronization test). This API does not return until the synchronization session ends, successfully or otherwise.

Restrictions:

Only one synchronization session can be active on the satellite at a time. If the db2SyncSatellite API is invoked and a synchronization session is currently active on the satellite, the following SQLCODE is issued:

```
SQL3950N A synchronization session is active.
At most one synchronization session can be active.
```

Procedure:

To initiate the synchronization session, use the db2SyncSatellite API as follows:

```
SQL_API_RC SQL_API_FN
db2SyncSatellite (      /* Synchronize the Satellite */
    db2UInt32 versionNumber, /* Database version number */
    void * pParmStruct,    /* Input/Output parameters */
    struct sqlca * pSqlca); /* SQLCA */
```

When this API ends successfully, the following SQLCODE is issued:

```
SQL3910I Synchronization session completed successfully.
```

Related reference:

- “db2SyncSatellite - Sync Satellite” in the *Administrative API Reference*

Querying Synchronization Session Progress Using the db2QuerySatelliteProgress API

The db2QuerySatelliteProgress API returns information about the progress of an active synchronization session.

If the db2QuerySatelliteProgress API successfully returns the progress information, the following SQLCODE is issued:

```
SQL3918I Synchronization progress information was obtained successfully.
```

If a synchronization session is not active when the API is invoked, the following SQLCODE is issued:

```
SQL3936W No progress information is available.
```

Procedure:

Use the db2QuerySatelliteProgress API as follows:

```
SQL_API_RC SQL_API_FN
db2QuerySatelliteProgress ( /* Query Satellite Progress */
    db2Uint32 versionNumber, /* Database version number */
    void * pParmStruct,      /* Output parameters */
    struct sqlca * pSqlca);  /* SQLCA */
```

Supply the following output parameters for the pParmStruct parameter structure:

```
typedef struct db2QuerySatelliteProgressStruct
{
    db2int32      oStep;          /* current step of synchronization */
    db2int32      oSubstep;       /* substep of the current step */
    db2int32      oNumSubsteps;   /* total number of substeps */
    db2int32      oScriptStep;    /* step of current script substep */
    db2int32      oNumScriptSteps; /* total number of script steps */
    char          *poDescription; /* description of step */
    char          *poError;       /* error text, if available */
    char          *poProgressLog; /* contents of progress log */
} db2QuerySatelliteProgressStruct;
```

Related reference:

- “db2QuerySatelliteProgress - Query Satellite Sync” in the *Administrative API Reference*

Stopping a Synchronization Session Using the db2SyncSatelliteStop API

The db2SyncSatelliteStop API can be used to interrupt an active synchronization session by issuing an asynchronous STOP request to the synchronization session. That is, the API returns immediately after it issues the STOP request, and not after the synchronization session stops. If a synchronization session is active and the STOP request is issued successfully, the following SQLCODE is issued:

```
SQL3912I STOP completed successfully.
```

If no synchronization session is active when the API is invoked, the following SQLCODE is issued:

```
SQL3913I STOP issued, but no synchronization session is currently active.
```

The db2SyncSatellite API only returns as a result of the interrupt when it has reached an appropriately *safe* point in the session. That is, a STOP request is not necessarily executed as soon as it is received. Instead, the db2SyncSatellite API finishes executing the current step in the synchronization process before exiting. For example, if the db2SyncSatelliteStop API is issued while the db2SyncSatellite API is executing a script, the db2SyncSatellite API completes executing the script before stopping. As another example, if the db2SyncSatellite API is interrupted before the satellite can upload the results of the synchronization session to the satellite control server, the API issues the following SQLCODE:

```
SQL3917I A STOP request was received before the results were uploaded to
the satellite control server.
The results will be uploaded during the next synchronization session.
```

The next time db2SyncSatellite is invoked, the synchronization session resumes where it stopped.

Procedure:

Use the db2SyncSatelliteStop API as follows:

```
SQL_API_RC SQL_API_FN
db2SyncSatelliteStop ( /* Stop synchronization session */
    db2UInt32 versionNumber, /* Database version number */
    void * pParmStruct, /* Input/Output parameters */
    struct sqlca * pSqlca); /* SQLCA */
```

You must execute the db2SyncSatellite API and the db2SyncSatelliteStop API on different threads or processes because the db2SyncSatellite API does not return until the synchronization session completes, and db2SyncSatelliteStop is asynchronous.

Related reference:

- “db2SyncSatellite - Sync Satellite” in the *Administrative API Reference*
- “db2SyncSatelliteStop - Stop Satellite Sync” in the *Administrative API Reference*

Building and Running Synchronizing Applications

The sections that follow describe how to build and run your own synchronizing application, and how to run the DB2 Synchronizer application.

Building and Running Your Synchronizing Application

You can build an application in any of the DB2-supported languages and application interfaces (APIs). However, the satellite APIs, required for a synchronizing application, are only available in C. This means that to build a

synchronizing application, either by programming the satellite APIs into your application, or calling them in a separate program, you need to program in C. You also need to build the synchronizing application on a supported Windows operating system with the Microsoft Visual C++ compiler.

The `db2sat.c` source file, located in the `sql11ib\samples\c` directory, is used to create the sample program, `db2sat`, which demonstrates satellite APIs for a synchronizing application.

Note: The DB2 Synchronizer is a far more comprehensive synchronizing application than the `db2sat` program. The `db2sat` program should mainly be used to learn how to program some of the main DB2 satellite APIs when developing your own application.

The batch file, `bldapp.bat`, in the same directory, contains the commands to build C application programs.

Procedure:

To build the sample program, `db2sat`, from the source file `db2sat.c`, enter:

```
bldapp db2sat
```

The result is an executable file, `db2sat.exe`. When the satellite environment is set up, you can run the executable file by entering the executable name (without the extension) on the command line:

```
db2sat
```

Related tasks:

- “Building C/C++ Applications on Windows” in the *Application Development Guide: Building and Running Applications*
- “Using the DB2 Synchronizer Application” on page 290

Using the DB2 Synchronizer Application

The DB2 Synchronizer is a synchronizing application that is available with all DB2 Universal Database products that can be used as satellites. The DB2 Synchronizer is available in both test and synchronize modes.

You use the DB2 Synchronizer in test mode to verify the ability of a satellite to synchronize with its satellite control server. You can also use the DB2 Synchronizer application to start a synchronization session if the application does not call the synchronizing APIs, or you have not written your own synchronizing application.

Procedure:

To verify the ability of a satellite to synchronize with its satellite control server:

1. Open the DB2 Synchronizer window in test mode by issuing the following command on the satellite:

```
db2sync -t
```

2. Start the synchronization test by clicking on the **Test** push button.

Notes:

- a. When a synchronization test is executed, the satellite does not download and execute batches.
- b. If the `satadmin.aut` file has been destroyed on a satellite, you can use the DB2 Synchronizer window in test mode to recreate it.

To start a synchronization session:

1. Open the DB2 Synchronizer window in synchronize mode by issuing the following command on the satellite:

```
db2sync
```

Note: If you performed the install interactively, the DB2 menu is available from the **Start** menu. If you installed DB2 interactively on the satellite, the DB2 menu is available from the **Start** menu. Click **Start** and select **Programs->IBM DB2->Set-up Tools->Satellite Synchronizer**.

2. Click on the **Start** push button.

Related concepts:

- “Synchronization Test Problems” on page 329

Related tasks:

- “Recreating or Updating the `satadmin.aut` File on a Satellite” on page 346

Chapter 13. Recovering the Satellite Environment

Recoverable Elements in a Satellite Environment	293	Refresh Recovery of the Model Office and Test Satellites	299
Recovering Control Information	294	Version Recovery of the Model Office and Test Satellites	301
Recovery of Control Information	294	Forward Recovery of the Model Office and Test Satellites	304
Recovering the Control Center Directories	295	Recovery of Satellites in the Production Environment	306
Recovering the Satellite Control Server and the Satellite Control Database	296		
Recovering the Test Environment	298		
Recovery of the Test Environment	298		
Recovery of the Model Office and Test Satellites	298		

The sections that follow describe how to recover the different parts of the satellite environment, and the strategies that you can use.

Recoverable Elements in a Satellite Environment

The following figure shows the different parts of the satellite solution. To ensure the availability of the satellite environment, you require recovery strategies for the control environment, and for the production environment. When you have decided on a recovery strategy for the satellite environment, you should test it to determine whether it meets all of your requirements.

Control Environment

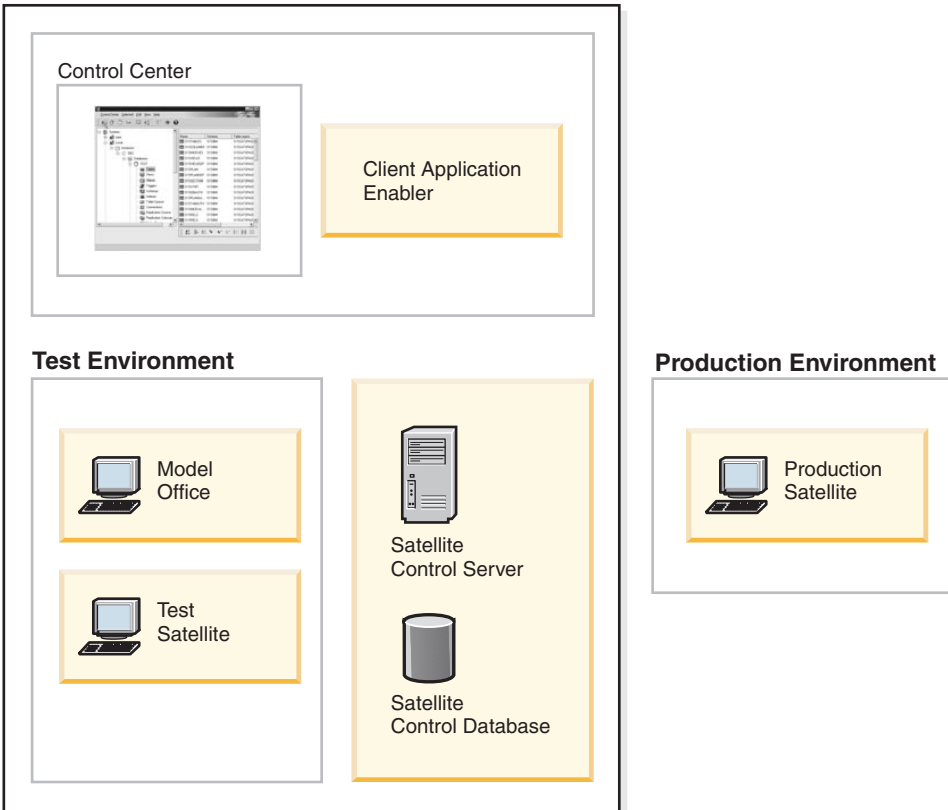


Figure 6. Recovery in the Satellite Environment

Related concepts:

- “Recovery of Control Information” on page 294
- “Recovery of Satellites in the Production Environment” on page 306

Recovering Control Information

The sections that follow describe how to recover control information.

Recovery of Control Information

The control segment of the satellite environment includes the information that you use to both set up and maintain the entire satellite environment. To be able to fully recover the control segment, you require backup images that you can use to restore the DB2[®] instances and the DB2 databases:

Instance-level considerations

In the control segment, the Control Center server (or the JDBC server),

the satellite control server, and the model office each has its own DB2 instance. To facilitate the restoration of any of these instances, you need to back up the database manager configuration file, and the node, database, and DCS directory of each of these three instances. You can perform this task by using the **db2cfexp** command with the **backup** parameter to export the file and the directories to a file. If you need to restore the database manager configuration and the communications information, you can easily do this by first recreating the instance, then using the **db2cfimp** command to import the file.

Database-level considerations

The control segment contains two databases: the satellite control database (SATCTLDB), which is on the satellite control server, and the database that you create on the model office. To be able to fully recover the satellite control database, you should enable it for forward recovery. In addition, you should also take a copy of the database configuration file `SQLnnnn\SQLDBCON`.

Whether the database on the model office is recoverable depends on the recovery strategy that you want to implement on your satellites.

Related concepts:

- “Recovery of the Test Environment” on page 298
- “Recovery of the Model Office and Test Satellites” on page 298

Related tasks:

- “Recovering the Control Center Directories” on page 295
- “Recovering the Satellite Control Server and the Satellite Control Database” on page 296

Related reference:

- “db2cfimp - Connectivity Configuration Import Tool Command” in the *Command Reference*
- “db2cfexp - Connectivity Configuration Export Tool Command” in the *Command Reference*

Recovering the Control Center Directories

When you specify the DB2 servers and DB2 databases that you want to use as execution targets by adding them to the Control Center, information about these DB2 servers and databases (such as the name, alias, and the host on which they reside) is stored in the node, database, and DCS directories of the Control Center instance. You can back up the information in these directories so that if you have to reinstall the Control Center, you do not have to recatalog all the execution targets.

Procedure:

Export the Control Center directories to a file using the **db2cfexp** command, with the **backup**, parameter as follows:

```
db2cfexp filename history
```

You should export these directories to a file every time you add a new DB2 server or database to the Control Center, and save the resulting file to a location other than the Control Center instance (for example, to a diskette).

If you need to re-install the Control Center, you can use the file to import its directories, and obtain the set up on the Control Center that you had before the re-installation. To import the directories from the file, you can use the Configuration Assistant. You can also use the **db2cfimp** command to import the file, as follows:

```
db2cfimp filename
```

Related tasks:

- “Importing a configuration profile : Configuration Assistant help” in the *Help: Configuration Assistant*

Related reference:

- “db2cfimp - Connectivity Configuration Import Tool Command” in the *Command Reference*
- “db2cfexp - Connectivity Configuration Export Tool Command” in the *Command Reference*

Recovering the Satellite Control Server and the Satellite Control Database

The availability of the satellite control server and the satellite control database is critical both to administering the satellite environment, and to the ability of satellites to synchronize. If the satellite control server or the satellite control database is not available, you cannot administer the satellite environment, nor can satellites synchronize.

To ensure the availability of the satellite control server, you should also consider setting up a high-availability solution for it.

Procedure:

For availability reasons, you should enable the satellite control database for forward recovery. To do this, you can either set the *logretain* database configuration parameter to Recovery or On, use a user exit to archive log files,

or both. With forward recovery, you will be able to reconstruct the satellite control database to either a specific point in time, or to the last update to the database that is recorded in the logs.

You should also keep a copy of the database configuration file `SQLnnnnn\SQLDBCON`. You should take a copy of this file every time that you change the value of one or more of the database configuration parameters. Changes to database configuration parameters are not recorded in the logs. If you maintain a copy of the database configuration file, you can restore the file after rolling the database forward. In this way, you do not have to use the `UPDATE DATABASE CONFIGURATION` command to update the database configuration after the roll-forward operation is complete. You will, however, have to stop and restart the database so that the configuration values specified in the configuration file are activated for the database.

To protect the satellite control database against media failure, you can use the `BACKUP DATABASE` command to write the backup image of the database to a different physical device than the device on which the database resides. You can also use the `newlogpath` database configuration parameter to specify that the logs required for forward recovery are stored on a different physical device than the database.

In addition to storing backup images and logs on a different device than the database, you can further protect against media failure by using disk mirroring or RAID.

Related concepts:

- “Rollforward Recovery” in the *Data Recovery and High Availability Guide and Reference*
- “Rollforward Overview” in the *Data Recovery and High Availability Guide and Reference*
- “High Availability” in the *Data Recovery and High Availability Guide and Reference*

Related reference:

- “Change the Database Log Path configuration parameter - newlogpath” in the *Administration Guide: Performance*
- “Log Retain Enable configuration parameter - logretain” in the *Administration Guide: Performance*
- “BACKUP DATABASE Command” in the *Command Reference*
- “UPDATE DATABASE CONFIGURATION Command” in the *Command Reference*

Recovering the Test Environment

The sections that follow describe how to recover the test environment.

Recovery of the Test Environment

The test environment typically contains one model office for each application version supported in the group, and the test satellites. You need to have a tested recovery strategy in place for both the model office, and for the test satellites.

Because the test satellite can be installed from the image that you create from the model office, the recovery strategy that applies to the test satellite is the same as that of the model office. That is, you will use refresh recovery, version recovery, or forward recovery to restore the database on the test satellite.

Related concepts:

- “Recovery of the Model Office and Test Satellites” on page 298
- “Refresh Recovery of the Model Office and Test Satellites” on page 299
- “Version Recovery of the Model Office and Test Satellites” on page 301
- “Forward Recovery of the Model Office and Test Satellites” on page 304

Recovery of the Model Office and Test Satellites

When you use a test satellite as your model office, it is the model of all the group satellites, both test and production. You need to be able to recover the model office for two reasons:

- The model office is representative of the group satellites. If you want to know the average state of the group, the model office must be available.
- Because the model office is representative of the group satellites, you can use it to derive changes in the database definition that is use in the group. For you to be able to use the model office to derive changes in the database definition, the model office must be available.

You also need to be able to recover your test satellites to ensure that you can fully test modifications to the group batches before you promote them to production.

An important decision to make when deciding on the configuration of the model office is the type of recovery that you want to use to restore the database. The recovery method that you decide to implement will apply to all the test and production satellites that are at the same application version as the model office.

Three types of database recovery are available. You can use refresh recovery, version recovery, or forward recovery to restore the database:

- With *refresh recovery*, you reinstall the model office or model office, using the install image that you used for the mass deployment. Refresh recovery is the easiest method to use to recover a database.
- With *version recovery*, you can restore the database to the state it was in when the backup image was taken. Version recovery is more expensive than refresh recovery, as you must maintain backup images of the database. In addition, you must be able to identify which backup image you want to use to restore the database.
- With *forward recovery*, you restore the database using a backup image, then use the logs to apply changes that are subsequent to the backup image. Forward recovery is the most expensive method of recovering a database, as you must maintain backup images of the database and the database logs. In addition, forward recovery can be the most error-prone method of recovering a database.

Related concepts:

- “Refresh Recovery of the Model Office and Test Satellites” on page 299
- “Version Recovery of the Model Office and Test Satellites” on page 301
- “Forward Recovery of the Model Office and Test Satellites” on page 304

Refresh Recovery of the Model Office and Test Satellites

With refresh recovery, you do not use DB2® utilities to either back up or restore the database on the model office or test satellite. Instead, you use the satellite install image to re-install the entire satellite, then have the satellite re-execute all its group batches, from the first batch step of each batch. In effect, with refresh recovery, you rebuild the model office or test satellite.

The method that you use to have the model office re-execute its group batches depends on whether the model office is configured as a test satellite or as a production satellite:

- If the model office is a test satellite, use the Edit Satellite notebook to set the model office to re-execute its group batches from the first batch step. (You also use the Edit Satellite notebook have the test satellite re-execute its group batches.)
- If the model office is a production satellite, use the Set Execution Starting Point window to set the model office to re-execute its group batches from the first batch step.

You can identify whether the model office is configured as a test or a production satellite from the satellite details view, which is available from the Satellite Administration Center.

Refresh recovery may be sufficient if none of your production satellites will contain unique data. That is, all of the data on the satellite is a replica of corporate data.

To use refresh recovery, set the *logretain* database configuration parameter to No. To set a database configuration parameter, you can use either the Configure Database notebook, which is available from the Control Center, or the UPDATE DATABASE CONFIGURATION command.

Model office development phase considerations:

Before you complete the development of the model office, you will likely not have an install image that you can use for refresh recovery. Instead, you may need to rebuild the model office by performing such tasks as reinstalling the operating system and DB2, and recreating the database definition. The information that follows provides some guidelines that may help simplify the process of rebuilding the model office.

If a problem occurs during the development phase, you can simplify setting up communication information on the model office by either:

- Using the **db2cfexp** command with the BACKUP parameter to export the node, database, and DCS directories of the model office. Then, after you re-install DB2, you can use the **db2cfimp** command to import the directories into the instance.
- Following the process described in Cataloging Remote Instances and Databases on the Model Office to recreate the node, database and DCS directories on the model office.

Consider keeping a copy of the database configuration file `SQLnnnnn\SQLDBCON` for the database on the model office. You can take a copy of this file every time that you change the value of one or more of the database configuration parameters. If you need to rebuild the model office, you can use the file to restore the database configuration without having to use the UPDATE DATABASE CONFIGURATION command. You have to stop and restart the database so that the configuration values specified in the configuration file are activated for the database.

If you can easily recreate the database and its data (for example, you have the SQL statements and DB2 commands in a file), you will likely not require a backup image of the database. If, however, this assumption is not true, consider backing up the database, and storing the backup image in a location other than the model office.

Related tasks:

- “Configuring a database: Control Center help” in the *Help: Control Center*

- “Editing a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Setting the execution starting point for a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Viewing satellite details : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Related reference:

- “Log Retain Enable configuration parameter - logretain” in the *Administration Guide: Performance*
- “UPDATE DATABASE CONFIGURATION Command” in the *Command Reference*
- “db2cfimp - Connectivity Configuration Import Tool Command” in the *Command Reference*
- “db2cfexp - Connectivity Configuration Export Tool Command” in the *Command Reference*
- “asncap: Starting Capture (UNIX, Windows, z/OS)” in the *Replication Guide and Reference*

Version Recovery of the Model Office and Test Satellites

With version recovery, you use DB2[®] utilities to take an offline backup image of the database on the model office or test satellite, which you can use to restore the database, as required. You have the option of taking a backup of the database every time you implement a change that results in the database being in a consistent state, or taking only one backup image of the database, then having the satellite re-execute its group batches.

You take an offline backup of the database using the BACKUP DATABASE command.

Note: For an offline backup to succeed, the backup database operation must be the only application that is connected to the database. To ensure that the backup operation is the only application that is connected to the database, you can use the FORCE APPLICATION ALL command or the DB2STOP FORCE command before issuing the BACKUP DATABASE command.

- If you want to have multiple backup images of the database, back it up when one of the following events occur. In either situation, the database is in a known and a consistent state:
 - You promote a batch to production to change the database definition that supports the application.
 - You successfully apply a fix batch to the model office or test satellite.

If you need to restore the database, you can restore it using the latest backup image that you have. Taking an offline backup each time that the database is in a consistent state as the result of a change to a batch or the successful application of a fix batch, however, can result in many backup images of the database.

- If you want to maintain only one backup copy of the database, take the backup image before the satellite has executed any of its group batches (that is, the satellite has not yet synchronized). If you need to restore the database, you can use this backup image, then have the satellite re-execute all of its group batches, from the first batch step of each batch.

The method that you use to have the model office re-execute its group batches depends on whether the model office is configured as a test satellite or as a production satellite:

- If the model office is a test satellite, use the Edit Satellite notebook to set the model office to re-execute its group batches from the first batch step. (You use the Edit Satellite notebook with test satellites.)
- If the model office is a production satellite, use the Set Execution Starting Point window to set the model office to re-execute its group batches from the first batch step.

You can identify whether the model office is configured as a test or a production satellite from the satellite details view, which is available from the Satellite Administration Center.

To use version recovery, set the *logretain* database configuration parameter to No. To set a database configuration parameter, you can use either the Configure Database notebook, which is available from the Control Center, or the UPDATE DATABASE CONFIGURATION command.

If you are using version recovery, you should export the database manager configuration file and the node, database, and DCS directories on the model office to a file. This way, if you need to restore the configuration on the model office, you can restore it by importing the file, then restoring the database backup. You can export the directories and database manager configuration to a file using the **db2cfexp** command with the BACKUP parameter.

With version recovery, you do not need to maintain a separate copy of the database configuration file *SQLnnnn\SQLDBCON*. When you back up the database, the configuration file is automatically saved with the backup image. If you always back up the database after making a change to the database definition, you should always be able to restore the database and its configuration. If you only maintain the initial backup image of the database, the database configuration will be updated when the satellite re-executes its group batches.

If you take a backup image every time the database is at a consistent state and you store the backup images on the satellite, disk space may be an issue. You should decide how many backups that you want to maintain, and consider using a fix batch to delete older backup images that you do not want to retain. Consider deleting the older backups only after the current backup completes; otherwise, if a backup operation fails, your recovery strategy may be compromised. You can use the LIST HISTORY command with the BACKUP parameter to return the list of backups that are available. When you delete a backup image, remember to use the PRUNE HISTORY command to remove the record for the backup from the history file.

If you want to protect the database against media failure on the model office, you should consider storing the backup image on a different physical disk than the disk on which the database resides.

If anything happens to the database on the model office, you can restore it to its last consistent state. For example, if the results of a group batch or a fix batch are not satisfactory, instead of applying a fix batch, you can restore the database from the previous backup image, and try the change again. When you obtain the results that you want, then take a backup of the database. To restore the database, you can use the Restore Database notebook or the Restore Database wizard, both of which are available from the Control Center. You can also use the RESTORE DATABASE command.

Related tasks:

- “Creating Execution Targets” on page 256
- “Backing Up and Restoring Databases Overview: Control Center help” in the *Help: Control Center*
- “Configuring a database: Control Center help” in the *Help: Control Center*
- “Editing a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Setting the execution starting point for a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Viewing satellite details : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Related reference:

- “Log Retain Enable configuration parameter - logretain” in the *Administration Guide: Performance*
- “BACKUP DATABASE Command” in the *Command Reference*
- “FORCE APPLICATION Command” in the *Command Reference*
- “RESTORE DATABASE Command” in the *Command Reference*
- “STOP DATABASE MANAGER Command” in the *Command Reference*

- “UPDATE DATABASE CONFIGURATION Command” in the *Command Reference*
- “LIST HISTORY Command” in the *Command Reference*
- “PRUNE HISTORY/LOGFILE Command” in the *Command Reference*
- “db2cfimp - Connectivity Configuration Import Tool Command” in the *Command Reference*
- “db2cfexp - Connectivity Configuration Export Tool Command” in the *Command Reference*
- “asncap: Starting Capture (UNIX, Windows, z/OS)” in the *Replication Guide and Reference*

Forward Recovery of the Model Office and Test Satellites

If you are using forward recovery, you can restore the database and apply the changes in the logs to roll the database forward either to a specific point in time, or to the end of the logs. In this way, you can recreate any configuration state of the model office or test satellite database.

You should only consider using forward recovery if your production satellites will contain unique data in the database that supports the application. That is, not all of the data in this database will be a replica of corporate data. Because you will not be able to recreate the data on a production satellite from corporate databases, you should retain the database logs if it is important that this unique data can be recovered. The most likely situation in which you would use forward recovery is to recover the database from an error that cannot be undone by a fix batch. In this situation, you would restore the database and roll it forward to the end of the logs to recover the unique data on the satellite.

To use forward recovery, set the *logretain* database configuration parameter to Recovery. The value of Recovery retains the logs that are required for forward recovery. To set a database configuration parameter, you can use either the Configure Database notebook, which is available from the Control Center, or the UPDATE DATABASE CONFIGURATION command.

If you are using forward recovery, you should export the database manager configuration file and the node, database, and DCS directories on the model office to a file. This way, if you need to restore the configuration on the model office or test satellite, you can restore it by importing the file, then restoring the database backup and rolling forward. You can export the directories and database manager configuration to a file using the **db2cfexp** command with the BACKUP parameter.

With forward recovery, you may not need to maintain a copy of the database configuration file `SQLnnnnn\SQLDBCON`. Whenever you back up the database, the

configuration file is saved with the backup image. If, however, the database configuration is changed and the database is not backed up when the change occurs, you should take a copy of the configuration file. After you roll the database forward, you can use the file to restore the database configuration without having to use the `UPDATE DATABASE CONFIGURATION` command. You have to stop and restart the database so that the configuration values specified in the configuration file are activated for the database.

If you are storing the backup images and logs on the model office or test satellite, disk space may be an issue. You should decide how many backups that you want to maintain, and consider using a batch step to delete the older backup images, as well as their associated logs. You can use the `LIST HISTORY` command with the `BACKUP` parameter to return the list of backups that are available. This command also displays the name of the first log file that is associated with the backup. Consider deleting the older backups and logs only after the current backup completes; otherwise, if a backup operation fails, your recovery strategy may be compromised. When you delete a backup image, remember to use the `PRUNE LOGFILE` to delete the logs associated with the backup image, and the `PRUNE HISTORY` command to remove the record for the backup from the history file.

If you want to protect the database against media failure, you should consider storing the backup image on a different physical disk than the disk on which the database resides. You should also store the logs on a different physical disk than that of the database. To perform this task, use the `newlogpath` database configuration parameter.

Related tasks:

- “Configuring a database: Control Center help” in the *Help: Control Center*

Related reference:

- “Change the Database Log Path configuration parameter - `newlogpath`” in the *Administration Guide: Performance*
- “Log Retain Enable configuration parameter - `logretain`” in the *Administration Guide: Performance*
- “`UPDATE DATABASE CONFIGURATION` Command” in the *Command Reference*
- “`LIST HISTORY` Command” in the *Command Reference*
- “`PRUNE HISTORY/LOGFILE` Command” in the *Command Reference*
- “`db2cfexp` - Connectivity Configuration Export Tool Command” in the *Command Reference*

Recovery of Satellites in the Production Environment

Because the model office is the template for the production satellites, the recovery strategy that you decide on for the model office also applies to the production satellites. The considerations for using refresh, version, or forward recovery on production satellites are as follows:

Refresh recovery

With refresh recovery, you do not use database backups to recover the database on the production satellite. Instead, you use the install image that was generated from the model office to re-install the entire satellite, then use the Set Execution Starting Point window to specify that the production satellite re-execute all of its group batches from the first batch step of each batch.

Version recovery

If you are using version recovery, you can back up the database on a periodic basis. In this situation, you will want to schedule how frequently the database is backed up. For example, you may want to take a backup image of the database on a weekly, or even a monthly basis. Or you may want the database to be backed up more frequently. One important factor to consider is the volume of transactions that are applied to the database. If the database is frequently updated, you may want to back up the database more frequently. With version recovery, the backup schedule will determine how much you can protect against the loss of data.

Forward recovery

If you are using forward recovery, you will want to schedule how frequently the database is backed up. The more frequent the backup, the fewer the logs that will have to be applied during the roll-forward phase when the database is recovered. For example, if updates to the database are infrequent, you may want to take a backup image of the database on a weekly, or even a monthly basis. Because few log records will be written, it will not require much time to fully recover the database. If, however, the database is updated frequently, you may want to back up the database more frequently to reduce the number of logs, and the amount of time that will be required to fully restore the database.

Related concepts:

- “Refresh Recovery of the Model Office and Test Satellites” on page 299
- “Version Recovery of the Model Office and Test Satellites” on page 301
- “Forward Recovery of the Model Office and Test Satellites” on page 304

Related tasks:

- “Setting the execution starting point for a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Chapter 14. Performing a Mass Deployment

Performing a Mass Deployment	309	Operating System Considerations for a	
How to Perform a Mass Installation.	310	Mass Copy	319
Performing a Mass Installation	310	Completing the Mass Deployment	320
Role of the Model Office in a Mass		Installing a New Version of the Application	
Installation	312	on Group Satellites	320
Customizing the Generated Response File		Installing a New Version of the	
for a Mass Installation	313	Application	320
Preparing and Using Your Distribution		Setting the New Application Version on	
Media for Mass Installation	314	the Satellite for the New Version of the	
Customizing the Operating Environment		Application	321
of Each Satellite During Mass Installation.	316	Creating and Testing the Group Batches	
Completion of the Satellite Setup During		for the New Version of the Application	322
a Mass Installation	317	Creating a Test System to Test the	
How to Perform a Mass Copy	317	Deployment of the New Application	323
Performing a Mass Copy	318	Deploying the New Application Version	
DB2 Considerations for a Mass Copy	318	to the Production Satellites.	325
Application Data Considerations for a		Monitoring Which Satellites Implemented	
Mass Copy	319	the New Application Version	325

The sections that follow describe how to perform a mass deployment of satellites. Also included is information on how to install a new version of an application on already-deployed satellites.

Performing a Mass Deployment

When you finish your development phase, and have verified that your model office represents what you want to deploy in production, you are ready to begin a mass deployment. Facilities exist to ease the deployment of the DB2 on hundreds and thousands of satellites.

If your deployment is large, it is *critical* that you test your deployment method rigorously before using it to create your entire production environment. Even after testing the method in the information systems test environment, consider deploying a set of pilot satellites before deploying a production environment.

Prerequisites:

Before performing a mass deployment, you should create your model office and use it to fully test the group batches of the application version.

Procedure:

You can use two methods to achieve the mass deployment of satellites:

- **Mass installation**

The mass installation method uses a traditional installation process to deploy DB2. Use this method if you cannot replace the contents of the entire hard disk on the satellite. This situation occurs:

- If the equipment is not owned by your company.
- When other applications are already deployed on company-owned systems.

For example, you are deploying a new application, the first one to use DB2, on systems that are already running other applications.

If you use the mass installation method, you must build your own customized installation application. Your application will have to invoke the DB2 installation application, and perform any additional customization that you require. You can use your installation application to install your application, in addition to DB2.

- **Mass copy**

When the equipment is company owned and dedicated to a single application, the setup of the application and database often occurs at a central site.

If you perform the setup activities at a central site, this can bypass the traditional installation process for all but the first satellite. In this situation, you will find it more efficient to copy the disk image of a fully configured system and perform minor customization, rather than have to repeatedly install the operating system, database and application. After you copy the disk, you can customize the satellite by running scripts that tailor the system, and, if required, load the data that is appropriate for that system.

Related tasks:

- “Performing a Mass Installation” on page 310
- “Performing a Mass Copy” on page 318
- “Creating and Testing Group Batches” on page 254

How to Perform a Mass Installation

The sections that follow describe how to achieve a mass deployment of satellites using mass installation.

Performing a Mass Installation

DB2 Universal Database provides tools that you can use to perform a mass installation. These tools include:

- A response file generator that you can use to reverse engineer the DB2 Universal Database products and components installed on a system to produce a response file. You can use the response file to drive the installation of DB2 on additional systems.
- Client profile export and import tools that you can use to move catalog information, ODBC/CLI settings, database manager configuration values, and DB2 registry settings from one system to other systems.
- A program, **cpyssetup.bat**, that you can use to copy the installation files for a specific language from the IBM distribution CD to your media.

The response file generator and the client profile exporter require a source system from which to extract the information that create their output files. You should use the model office as the source system for these utilities.

Procedure:

When you combine the output from the utilities described above with your custom installation application, you can create the media to support a mass installation. Use the following figure as a guideline as to how to perform a mass installation.

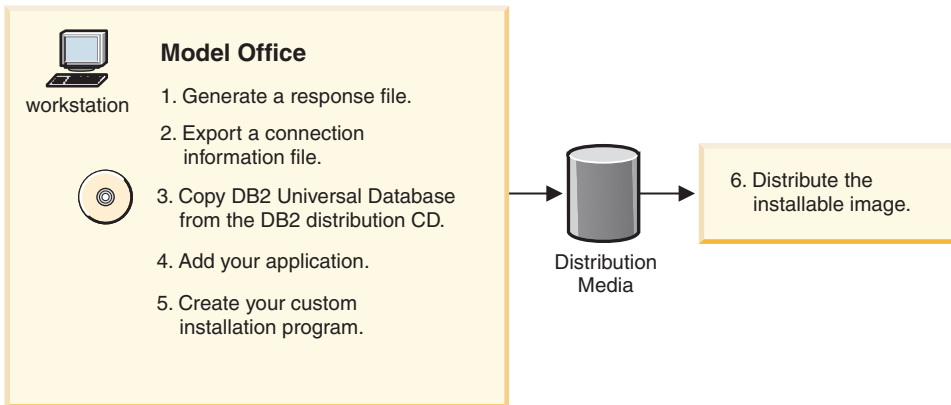


Figure 7. A Mass Installation

Related concepts:

- “Role of the Model Office in a Mass Installation” on page 312
- “Completion of the Satellite Setup During a Mass Installation” on page 317

Related tasks:

- “Customizing the Generated Response File for a Mass Installation” on page 313

- “Preparing and Using Your Distribution Media for Mass Installation” on page 314
- “Customizing the Operating Environment of Each Satellite During Mass Installation” on page 316

Role of the Model Office in a Mass Installation

You can use the model office during the production-deployment phase as the source system for the utilities that can generate:

- Installed component information and other parameters that are required by the installation program
- Catalog information
- ODBC/CLI settings, database manager configuration values, and DB2® registry settings.

When the model office is set up and tested to ensure that it represents the production environment for a specific version of the application, you can use the response file generator utility, **db2rspgn**, to generate a response file that you can use to install your production satellites. In addition to generating a response file that can be used to perform installations, the response file generator calls the client profile export utility to generate a client profile file for each instance that contains:

- System-level values for DB2 registry settings
- Global CLI/ODBC settings
- Instance-level database manager configuration setting, DB2 registry settings, and catalog information for node and database directory entries
- CLI/ODBC settings for CLI/ODBC enabled databases.

Note: In most situations, only the DB2 instance will exist on the model office. One client profile will be generated for it.

The generated response file contains keywords that reference the generated client profile files. When the response file is used during a response-file driven installation, the installation program calls the client profile import utility so that the contents of the referenced client profile files can be imported to the new satellite to set it up. Using this method, you can replicate the setup of the model office to hundreds of satellites.

Related concepts:

- “Response files” on page 114

Related tasks:

- “Creating a response file on Windows” on page 132
- “Response file installation of DB2 on Windows” on page 129

- “Customizing the Generated Response File for a Mass Installation” on page 313

Customizing the Generated Response File for a Mass Installation

The response file that the **db2rspgn** utility generates cannot extract all of the information that is required to install DB2 from the model office. For example, the password for the user name that is used by the Remote Command Service cannot be extracted from Windows NT. So that you can easily specify this additional information, the generated file contains sample keywords. You must update some of these keywords before using the response file for the installation.

Prerequisites:

You must have created the generated response file from the model office.

Procedure:

When you set the values in the response file, consider the following:

- Satellite environment specific keywords:
 - The satellite ID must be unique on each satellite system. If the satellite ID is not set, the logon ID of the user of the satellite is used as the satellite ID. If you decide to set the satellite ID using a keyword in the response file, the satellite ID will be the same on all the satellites that are installed using the response file, unless you customize the response file for each satellite before it is used.
 - You must set the user ID and password for the Remote Command Service using the `ADMIN.USERID` and `ADMIN.PASSWORD` keywords.
 - You can set the user ID and password that the satellite uses to connect to the satellite control database for synchronization.
- General DB2 installation keywords:
 - In most situations, you will want to set `REBOOT = NO` to prevent the DB2 installation program from rebooting the computer when the installation of DB2 completes. If the computer is not rebooted, your installation application, which launched the DB2 installation program, will regain control when the DB2 installation completes. In this way, more customization of the satellite can be done. If you do not reboot the computer, be aware of the following:
 - All services added by the DB2 installation program will not be started.
 - All running processes will not have access to the updated path information that is created by the DB2 installation program. If you want to run any DB2 commands from your installation application

after the DB2 installation completes, you must explicitly code the path information for all DB2 commands that you issue before the reboot.

- Set `DB2.AUTOSTART = NO` if you do not want the DB2 instance to be started when the computer is rebooted.
- Do not add the following keywords to the generated response file:
 - `DB2.PORT_NUMBER`
 - `DB2.SVCENAME`

Their values are overwritten when the DB2 instance client profile is imported.

You may not be able to specify in the response file where on the computer you want DB2 to be installed. In this situation, your installation application can prompt for the location in which to install your application and DB2. With this information, your install application can edit the response file and replace the value of the `FILE` keyword with the drive and path in which DB2 is to be stored before the DB2 installation is performed. Similarly, you can change the values of other keywords before installing DB2.

If you are modifying the response file, you must first copy it to a disk drive on the target satellite before editing the file. If you copy the response file to a disk drive, copy the generated client profile files to the same drive and path. You must do this because the DB2 installation program only searches the directory location of the response file is for client profile files.

Related tasks:

- “Customizing the Operating Environment of Each Satellite During Mass Installation” on page 316

Related reference:

- “Response file keywords” on page 116
- “db2rspgn - Response file generator” on page 124

Preparing and Using Your Distribution Media for Mass Installation

When you prepare your distribution media, the layout of the files that will make up the various components of the DB2 installation image should be considered. For example, assume that you are preparing the files to install DB2 so that they can be used to burn a CD. Further assume that all the files required for the installation of DB2 will be placed on the E drive of your system in a directory called `\satellite`. Into this directory, you will need to copy the generated and customized response file, the generated client profiles, and the files required to install DB2.

Procedure:

To prepare and use your distribution media for a mass installation:

1. Copy the DB2 installation files to your media.

The **cpyssetup.bat** utility is provided on the DB2 distribution CD. You use this utility to copy the files that are required to install DB2 to your distribution media.

Assume that your CD-ROM drive is the G drive. To copy the DB2 installation files to the e:\satellite directory, you would use the following command:

```
g:\db2\common\cpyssetup.bat e:\satellite EN
```

The resulting file structure would be as follows:

```
e:\satellite\setup.exe
      \db2\common
```

2. Copy the generated response file and client profile file to your media.

When you copy the generated and customized response file and client profile files to your media, they must be in the same directory. If they are not, the DB2 installation program cannot locate the client profile files and import them.

For the example shown in step 3, it is assumed that you copied these files into a subdirectory of e:\satellite\response.

Note: If you want, the response file and the client profile files can be generated directly to the image of your distribution media.

3. Invoke the DB2 installation program from your installation application.

To invoke the DB2 installation program from your application, install using a response file. In the example that has been used, if the install image was built on the E: drive, the command to run the installation would be as follows (the command is over two lines for formatting reasons):

```
e:\satellite\setup /u e:\satellite\response\db2udbse.rsp
      /l drive:\path\logfile
```

You must set the installation log file location appropriately for use on the target system.

If you require additional input to further customize the response file, the response file and the generated client profile files must have been copied to the disk drive on the target system. In this situation, you must modify the **setup** command to point to the location of the updated response file.

Related tasks:

- “Response file installation of DB2 on Windows” on page 129

Customizing the Operating Environment of Each Satellite During Mass Installation

Each satellite has DB2 configuration values that must be set up so that the satellite can operate in the satellite administration environment.

Procedure:

During a mass installation, use your installation application to set or modify the configuration values:

- The satellite ID, user ID, and password required for authentication with the satellite control server.

The satellite ID must be unique on each satellite, but the ID does not have to be explicitly set. If the satellite ID is not set on the satellite, the logon ID is used to uniquely identify the satellite.

The satellite ID is determined in the same way when the `satadmin.aut` file is created to contain the authentication credentials that are required to connect to the satellite control database. If the satellite ID is not set, the user ID of the individual who installs DB2 is used to create the entry in the `satadmin.aut` file. If the satellite ID is subsequently set on the satellite, or the user who operates the synchronizing application is not the user who installed the satellite, the authentication credentials required for connecting to the satellite control database will not be found. You can use one of the following methods to fix this problem:

- Do not set the `SATELLITE_ID` keyword, or provide the user ID (`DB2.SATCTLDB_USERNAME`) and password (`DB2.SATCTLDB_PASSWORD`) that are required to connect to the satellite control database in the response file:
 - You can have your installation application set the `SATELLITE_ID` keyword by calling the `db2set` command after the installation of DB2 completes.
 - Before the satellite synchronizes, a synchronization test (`db2sync -t`) must be run on the satellite. During the synchronization test, the user ID and password that are required for synchronization must be specified. When this is done, the authentication credential (user ID and password) is stored under the `SATELLITE_ID` in the `satadmin.aut` file and is for subsequent synchronizations.
- Provide both the `SATELLITE_ID`, and the user ID and password required to connect to the satellite control database in the response file. Because the satellite ID must be unique on every satellite, you must customize the response file to set a unique value before the DB2 installation program is started. To perform this task, copy the response file to a disk drive on the satellite, and use a program to edit the file to set a unique value for `DB2.SATELLITE_ID`. If you decide to use this method, ensure that you:

- Copy the client profile files into the same directory as the response file; otherwise, the DB2 installation program cannot find them
- Modify or dynamically generate the command that invokes the DB2 setup program to set the drive and path where the modified response file is stored.

Note: The satellite ID must be unique on each satellite. To prevent the ID from being overwritten when the client profile files are imported at the end of the installation process, the ID is not included in the exported client profile files.

- Satellite application version

The value that is on the model office is exported when the client profile file is created. This value is imported when the client profile is imported. You can set the value again, either after DB2 is installed, or when the application is installed on the satellite. You can use either the **db2sync -s** command or the `db2SetSyncSession` API.

Related tasks:

- “Setting the DB2SATELLITEID Registry Variable on a Satellite” on page 250

Related reference:

- “db2sync - Start DB2 Synchronizer Command” in the *Command Reference*
- “db2SetSyncSession - Set Satellite Sync Session” in the *Administrative API Reference*

Completion of the Satellite Setup During a Mass Installation

After you have installed and customized DB2® on the satellite, you will need to install the application software and create the database, and any database objects that are required to support the application (such as tables and indexes). You also need to bind the application to the database, unless it only uses CLI and ODBC commands to access data. You can accomplish these tasks as part of your custom installation program, or you can perform a second installation.

Related tasks:

- “Completing the Mass Deployment” on page 320

How to Perform a Mass Copy

The sections that follow describe how to achieve a mass deployment of satellites using mass copy.

Performing a Mass Copy

Mass copy is another method that you can use to achieve a mass deployment of satellites.

Prerequisites:

For this mass copy method to be suitable, the following must be true:

- The model office that you copy is operational. That is, it can synchronize.
- The model office is already customized for the satellite administration environment.

Procedure:

To accomplish a mass deployment using the mass copy method, copy the entire hard drive on the model office, using a copy utility such as Ghost. When the hard drive is copied, you will have to customize the resulting image before using it. For example, the satellite ID on the satellite will have to be set to a unique value, as will many operating system values, such as the TCP/IP host name.

To create the image, you can use the model office, or a copy of it.

After you copy the model office, you will have to perform customization to represent the unique environment that is required on each satellite (for example, customizing DB2 and operating system values).

Related concepts:

- “DB2 Considerations for a Mass Copy” on page 318
- “Application Data Considerations for a Mass Copy” on page 319
- “Operating System Considerations for a Mass Copy” on page 319

DB2 Considerations for a Mass Copy

If you perform a mass copy, you need to customize the following for DB2:

- General DB2[®] configuration:
 - Set the DB2SYSTEM registry variable to either the computer name or the TCP/IP host name of the system. DB2SYSTEM is a global DB2 registry variable, which you can set with the **db2set -g** command.
 - If the hardware you are deploying on and the model office system from which you generated the copy have significantly different CPU speeds, set the *cpuspeed* database manager configuration parameter to -1.
 - You can modify other DB2 registry values and database manager configuration values as required.

- Satellite specific configuration:
 - If you do not want to use the logon ID as the satellite ID, the DB2SATELLITEID registry variable must be set to a value that is unique in the satellite control database. When you do this, the authentication credentials in the satadmin.aut file that are required to connect to the satellite control database will no longer be valid. Before the satellite can be synchronized, a synchronization test (**db2sync -t**) must be run on the satellite. During the synchronization test, specify the user ID and password that are required for synchronization.
 - If the model office that is the source of the copied image already has your business application installed on it, the application version that is required for synchronization is most likely set. If, however, the application version is not set, specify the value by using either the **db2sync -s** command or the db2SetSyncSession API.

Related reference:

- “CPU Speed configuration parameter - cpuspeed” in the *Administration Guide: Performance*
- “db2set - DB2 Profile Registry Command” in the *Command Reference*
- “db2sync - Start DB2 Synchronizer Command” in the *Command Reference*
- “db2SetSyncSession - Set Satellite Sync Session” in the *Administrative API Reference*

Application Data Considerations for a Mass Copy

When you copy the hard drive of the model office, you not only copy the database and the definitions for tables, indexes, and other database objects, you also copy the data. Most likely, this data will not be the right subset of the corporate data for the satellite. You will have to delete and replace the data. You can populate the tables on the satellite with the load or import utilities.

Operating System Considerations for a Mass Copy

You will have to customize unique operating system values (such as the computer name) and communications information (such as the TCP/IP host name) for each satellite. The techniques that you use to accomplish the customization, and the list of characteristics that must be customized are beyond the scope of this topic.

When you have customized these characteristics, you are ready to complete the mass deployment.

Related concepts:

- Appendix F, “Workstation naming rules” on page 377

Related tasks:

- “Completing the Mass Deployment” on page 320

Completing the Mass Deployment

Before your deployed satellites can synchronize, you must define and set them up in the satellite control server. Assuming that you have been using the model office and test satellites to prepare for the mass deployment, the group, application version, and group batches for the application version already exist. You only need to add the deployed satellites to the group and promote the batches to production

Procedure:

In the Satellite Administration Center, locate the group for which the mass deployment is being performed, and using the Create Satellite notebook, add the satellites. You must also enable the satellites to execute group batches.

You should also ensure that the group batches are promoted to production so that the deployed production satellites can execute these batches when they synchronize.

Related tasks:

- “Promoting a Test Level to a Production Level in an Application Version” on page 214
- “Creating a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Promoting a Batch to Production : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Enabling all satellites in a group : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Enabling a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Installing a New Version of the Application on Group Satellites

The sections that follow describe how to install a new version of the application on group satellites.

Installing a New Version of the Application

During the life of your business application, you may find it necessary to upgrade the application to address changing business needs. You can upgrade the application to a new version by:

- Installing DB2. The installation can be either the initial installation of DB2, or an upgrade to a new release or version.
To install the application at the same time you install DB2, you can use either the mass install or the mass copy methods.
- Installing only the application, without any change to the level of DB2 on the satellite.

Procedure:

When you install a new version of the application:

1. Set the application version on the satellite to support the new application.
2. Create the new application version and the group batches that will set up the database definition required by the new application.
3. Create a test system to test the deployment of the new application.
4. Deploy the new application to the production satellites.
5. Optional. Monitor which satellites have implemented the new application version.

Related tasks:

- “Setting the New Application Version on the Satellite for the New Version of the Application” on page 321
- “Creating and Testing the Group Batches for the New Version of the Application” on page 322
- “Creating a Test System to Test the Deployment of the New Application” on page 323
- “Deploying the New Application Version to the Production Satellites” on page 325
- “Monitoring Which Satellites Implemented the New Application Version” on page 325
- “Performing a Mass Installation” on page 310
- “Performing a Mass Copy” on page 318

Setting the New Application Version on the Satellite for the New Version of the Application

A new version of an application often requires changes to the existing database definition. In addition to installing the new application code, you can use the installation application to perform any database definition migration and data migration that you require to support the new application. An alternative that you can also use is to drop the existing database, create a new database, and reload the required data.

When the new version of the application is installed on the satellite, you will not want it to execute the group batches of the old application version. If the satellite executes the batches for the previous application version, errors will likely occur. When your installation application installs the new version of the application on the satellite, it should also update the application version that is used by the satellite for the purpose of synchronizing.

Procedure:

Specify the new application version by using either the **db2sync -s** command or the `db2SetSyncSession` API.

If the update of application version is not performed before the satellite next synchronizes, the satellite will upload its old application version to the satellite control server, and any unexecuted batch steps for this application version will be sent to the satellite and executed. If any script in the group batches is incompatible with the new database definition, the results of the execution are unpredictable. For this reason, you must ensure that the application version is updated on the satellite *before* the synchronization session is initiated. Using the methods described above, reset the application version when you install the new version of the application.

Related reference:

- “db2sync - Start DB2 Synchronizer Command” in the *Command Reference*
- “db2SetSyncSession - Set Satellite Sync Session” in the *Administrative API Reference*

Creating and Testing the Group Batches for the New Version of the Application

To support the new version of the application on the group satellites, you must set up information about the group’s new application version on the satellite control server.

Procedure:

When you want to deploy a new version of the application, you must:

- Create the group batches that you require for the new application version. The satellites will execute these batches the first time that they synchronize after the new version of the application is installed. Initially, these batches should be at the test level (that is, you should not promote them until you are sure that they produce the results that you want).

Note: When you create the application version, ensure that it matches the new version of the application that you are deploying.

- Create a test system to test the deployment of the new application.

Related tasks:

- “Creating and Testing Group Batches” on page 254
- “Creating a Test System to Test the Deployment of the New Application” on page 323

Creating a Test System to Test the Deployment of the New Application

Before you deploy the new version of the application, you should create a test system that is at the same level as the existing production satellites. You can use this system to verify that the installation of the new version of the application provides the correct results, and that the setup, update and cleanup batches produce the expected database definition for the application.

Procedure:

To create a test system:

1. Use the installation process that you used to create the production satellites of the previous application version to install the test system. (You would have either performed a mass install or performed a mass copy.) The installation will set up the system to the level that was first deployed; that is, before the production satellites executed any of their group batches.
2. Add the test system to the group by using the Satellite Administration Center. You should name this test system using the same conventions that you are using for your model offices. When you accomplish a clean test, you will most likely want to use this system as the model office for the new application version within the group.
3. Have the system synchronize so that it runs any batches that are already in production for the previous version of the application:
 - Use the Satellite Administration Center to configure this satellite as a production satellite.
 - Enable the satellite to execute its batches.
 - Run the **db2sync** application on the satellite. The satellite will download and execute all batches and batch steps that are in production for the previous application version.
 - Check the results of the synchronization to ensure that the satellite is not in the failed state, and that the batches were executed as expected.
 - If the results are as expected, use the Satellite Administration Center to configure the system as a test satellite.

When you are finished this step, the test system is at the same level as the production satellites that are running the previous application version.

4. Install the new version of the application, using the process and media that you intend to use to upgrade your production satellites. You should use the installation process to accomplish any database migration and data migration steps.
5. Ensure that the installation program performs its task correctly:
 - If the database definition changed for the new application version, or data was migrated, verify that the results are as expected
 - Verify that the new application version is set on the satellite. You can use the **db2sync -g** command to view the application version.
6. Issue the **db2sync** command on the new model office so that it executes the batches that you created for the new application version.
7. Check the results:
 - You should answer the following questions:
 - Did the batches run successfully?
 - Is the satellite in the failed state?
 - Are there any unexpected results in the batch output logs?
 - If there are no errors, is the resulting database definition and data (if applicable) correct?
 - Correct any problems that occur, and repeat the synchronization test

Repeat the process of synchronizing, checking results, and debugging the group batches until the satellite can synchronize without errors. The process should be able to set up the test satellite correctly before you deploy your production satellites.

When you are ready to begin deploying the production satellites, promote the setup, update and cleanup batches to the production level (because this is a new application version, you promote test level 0).

When you use the procedure described above and can build a test system that runs without errors, you should retain it, and use it as your model office for the new application version. The system is already upgraded with the new version of the application, and has already executed the group batches that you promoted to production.

Related tasks:

- “Performing a Mass Installation” on page 310
- “Performing a Mass Copy” on page 318
- “Promoting the Batches of Test Level 0 to Production” on page 267
- “Identifying and Fixing a Failed Satellite” on page 337
- “Fixing Problems Caused by Test-Level Group Batches” on page 265

- “Creating a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Enabling a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Configuring a satellite as a production satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Configuring a satellite as a test satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Related reference:

- “db2sync - Start DB2 Synchronizer Command” in the *Command Reference*

Deploying the New Application Version to the Production Satellites

After you have validated the batches of the new application version by using a test system, you can deploy the new application version to the production satellites.

Procedure:

Install the new application version on the production satellites, using the process and media that you tested with the test system. Using the same process should set the new application version on each satellite.

The next time that the production satellite synchronizes, it will upload its new application version to the satellite control server, and download the batches for this new application version. The satellite will execute the batches, and upload the results of the synchronization to the satellite control server. If you want, you can monitor which satellites have implemented to new version of the application.

Related tasks:

- “Creating a Test System to Test the Deployment of the New Application” on page 323
- “Monitoring Which Satellites Implemented the New Application Version” on page 325

Monitoring Which Satellites Implemented the New Application Version

You can use the satellite details view in the Satellite Administration Center to determine which application version a satellite is running.

Procedure:

On the satellite details view, filter on the **Application version** column to determine which satellites have installed the new version, and which satellites are not yet upgraded.

Related tasks:

- “Viewing satellite details : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Chapter 15. Problem Determination

Installation Problems	327	Satellite's Application Version Does Not Exist at the Satellite Control Server.	334
Location of Error Messages for Satellite Control Server Installation	328	Application Version Is Not Set at the Satellite	335
Location of Error Messages for Satellite Installation	328	Authentication Error Occurs at Satellite Rebinding the db2satcs Stored Procedure Because of SQL0818	336
Configuration Problems That Prevent Synchronization	329	Identifying and Fixing Failed Satellites.	336
Synchronization Problems During Test Synchronizations	329	Identifying and Fixing a Failed Satellite	337
Synchronization Test Problems	329	Identifying the Failed Satellite	337
Satellite ID is Not Set on Satellite	329	Obtaining Information About the Failure on the Satellite	339
Satellite Control Database Not Cataloged on Satellite	330	Assigning a Fix Batch to the Satellite	340
Authentication Credentials Are Not Available Or Not Correct on the Satellite	331	Debugging the Fix Batch	342
Satellite Does Not Exist at the Satellite Control Database	331	Returning the Repaired Satellite to Production	343
Incompatible Versions of DB2 on the Satellite and the Satellite Control Server.	332	Running the DB2 Trace Facility on a Satellite	344
Synchronization Problems	332	Satellite Software Version	345
Resolving the Problem when a Satellite Can Only Synchronize Once	332	Internal and External Error Return Codes for Batch Steps	345
Satellite Is in the Failed State at the Satellite Control Server	333	Satellite Progress File	346
Satellite Is Not Enabled at the Satellite Control Server	333	Recreating or Updating the satadmin.aut File on a Satellite	346
		Determining Synchronization Errors when Log Details Are Truncated	347

The sections that follow describe some of the problems that can occur in the satellite environment. Topics that are covered include installation errors, and the configuration errors that can prevent a satellite from synchronizing. In addition, an overview is provided of how to fix a satellite that reports an error during a synchronization session.

Installation Problems

During the installation of either the satellite control server or DB2 on a satellite, errors can occur. Information that you can use to diagnose problems is logged in several locations. The sections that follow provide more detailed information.

Location of Error Messages for Satellite Control Server Installation

You install the satellite control server by selecting the Satellite Control Server component of DB2 Universal Database Enterprise Server Edition. Installation-related messages are logged as follows:

- On Windows-based platforms:
 - If you perform an interactive installation or a response file installation without redirecting the installation messages to a file using the /L option, the installation messages are written to the db2.log file as follows:

```
x:\db2log\db2.log
```

Where *x*: is the drive from which Windows® is booted (the boot drive).

- If you perform a response file installation using the /L option, the installation messages are written to the path and file that you specify.

You can use any editor to examine the file that is created during the installation process.

When the installation program creates the satellite control database, SATCTLDB, messages from this operation are logged in the *db2installpath\misc\satctl.db.log* file.

- On AIX® platforms, the installation messages are written to the /temp/db2setup.log file. You can examine this file using any editor. On AIX, you create the satellite control database, SATCTLDB, after installing the satellite control server. The messages for the create database operation are written to the log file that you specify.

Location of Error Messages for Satellite Installation

Installation messages for DB2® are logged as follows on Windows-based platforms:

- If you perform an interactive installation or a response file installation without redirecting the installation messages to a file using the /L option, the installation messages are written to the db2.log file as follows:

```
x:\db2log\db2.log
```

Where *x*: is the drive from which Windows is booted (the boot drive).

- If you perform a response file install using the /L option, install messages are written to the path and file you specify.

You can use any editor to examine the file that is created during the installation process.

If you perform a response file installation, and the response file imports client profile files to configure the database manager instance, error messages from the import process are written to the db2diag.log for each instance.

Configuration Problems That Prevent Synchronization

The sections that follow describe the configuration problems that can prevent either a test of the synchronization capability, or synchronization, from occurring. In general, the problems are caused by errors in the satellite configuration, information that is either missing or incorrect in the satellite control database, and authentication errors.

Synchronization Problems During Test Synchronizations

The sections that follow describe the problems that can occur during a synchronization test, and how to resolve these problems.

Synchronization Test Problems

You should use the **db2sync -t** command to open the DB2[®] Synchronizer application in test mode. You use the test mode to verify that the information that is required for the satellite to synchronize itself is correct both on the satellite, and in the satellite control database.

Related concepts:

- “Satellite ID is Not Set on Satellite” on page 329
- “Satellite Control Database Not Cataloged on Satellite” on page 330
- “Authentication Credentials Are Not Available Or Not Correct on the Satellite” on page 331
- “Satellite Does Not Exist at the Satellite Control Database” on page 331

Related reference:

- “db2sync - Start DB2 Synchronizer Command” in the *Command Reference*

Satellite ID is Not Set on Satellite

Typically, the satellite ID is the same as the logon ID for the satellite. If this is not the case, you can set the value of the DB2SATELLITEID registry variable on the satellite to define the satellite ID. In either situation, the satellite ID on the satellite must be equal to the value specified for the satellite ID when it is created in the Satellite Administration Center using the Create Satellite notebook.

When a synchronization session is initiated on the satellite, the satellite ID is determined as follows:

1. If a value is specified for the DB2SATELLITEID registry variable, the satellite ID is this value.
2. Otherwise, the logon ID is used as the satellite ID.

If the satellite ID cannot be determined when the synchronization session starts, the SQLCODE -3951N is returned. This error can occur if the DB2SATELLITEID registry variable is not set *and* no one has logged on to the satellite.

Depending on which method you use to specify the satellite ID, correct the problem as follows.

- If you use the logon ID as the satellite ID, ensure that the the person who initiates the synchronization session logs on using the user ID that is defined in the satellite control database as the satellite ID.
- If you use the DB2SATELLITEID registry variable to record the satellite ID, set the registry variable locally on the satellite as follows:

```
db2set DB2SATELLITEID=satellite_id
```

Notes:

1. The value specified for *satellite_id* must be equal to the value that is recorded as the satellite ID in the satellite control database. You can use the Satellite Administration Center to view the satellite ID from either the satellite details view, or the Edit Satellite notebook.
2. The satellite ID is case sensitive.

Related tasks:

- “Setting the DB2SATELLITEID Registry Variable on a Satellite” on page 250
- “Creating a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Editing a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Viewing satellite details : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Related reference:

- “db2set - DB2 Profile Registry Command” in the *Command Reference*

Satellite Control Database Not Cataloged on Satellite

Before the synchronization test can occur, the satellite control database must be recorded in the database directory on the satellite. If the information about the satellite control database is either not available on the satellite, or is not correct, the SQLCODE -3955 is returned. In this situation, the Catalog Control Database window opens, in which you can specify the DB2® instance that

contains the satellite control database. To specify the information, you can either use Discovery or type the TCP/IP host name and port number. For information about using the Catalog Control Database window, refer to the online help that is available from the window.

Authentication Credentials Are Not Available Or Not Correct on the Satellite

Before the synchronization test can occur, valid authentication credentials for the satellite control database must exist on the satellite and be stored in the *instance_path*\security\satadmin.aut file. If a synchronization test is initiated and either the satadmin.aut file does not exist, or it does not contain the user ID and password required to connect to the satellite control database, the SQLCODE -3966 is returned with a reason code of 1. In this situation, the Connect to Control Database window opens, which you use to specify the authentication credentials to use to connect to the satellite control database. For information about using the Connect to Control Database window, refer to the online help that is available from the window.

Satellite Does Not Exist at the Satellite Control Database

During a synchronization test, the satellite uploads its unique satellite ID to the satellite control server. The satellite control server then checks that this value is recorded in the satellite control database. If the satellite ID is not recorded in the satellite control database, the SQLCODE -3931W is returned. The synchronization test ends.

Note: The satellite ID is displayed in the title bar of the DB2[®] Synchronizer application.

If this error occurs:

- The satellite ID that the satellite uploads to the satellite control server is not correct.

In this situation, use the **db2set** command to change the value of the DB2SATELLITEID registry variable, or ensure that the individual who initiates the synchronization session is logged on to the user ID that corresponds to the satellite ID that is recorded in the satellite control database.

- The satellite ID is not recorded correctly in the satellite control database. In this situation, you must drop the satellite and create it again.
- The satellite is not yet created.

If the satellite is not yet created, use the Create Satellite notebook to create it.

Note: The satellite ID is case sensitive.

Related tasks:

- “Setting the DB2SATELLITEID Registry Variable on a Satellite” on page 250
- “Creating a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Deleting a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Related reference:

- “db2set - DB2 Profile Registry Command” in the *Command Reference*

Incompatible Versions of DB2 on the Satellite and the Satellite Control Server

For synchronization to occur, the release level of DB2[®] on the satellite control server and the release level of DB2 on the satellite must be compatible. The release level of DB2 on the satellite must be within the range of one level above to two levels below that of the DB2 on the satellite control server. If the release levels are not compatible, the SQLCODE -3933W is returned.

If this error occurs, migrate the release level of DB2 that is on the satellite so that it is compatible with the release level of DB2 on the satellite control server.

Note: A Version 8 satellite control server only supports satellites that are running DB2 Universal Database, Version 8.

Related concepts:

- “Satellite Software Version” on page 345

Synchronization Problems

The sections that follow describe problems that can occur during a synchronization session.

Resolving the Problem when a Satellite Can Only Synchronize Once

The situation can occur that, after encountering an error during a synchronization session and subsequently being fixed, the satellite cannot synchronize again. What can occur is as follows. The satellite synchronized, successfully downloading its group batches, but encountered an error when executing these batches. After being fixed, the satellite attempted to synchronize again, and received the SQLCODE -3950, indicating that a synchronization session is already active. This SQLCODE is issued because the satellite control server records the satellite as having downloaded its group batches, and is expecting the satellite to upload the results of the

previous synchronization session. The fix to the satellite, however, has set the state on the satellite so that the satellite is again attempting to download its group batches.

Procedure:

To fix this problem, you must change the status of the satellite in the satellite control database to enable it to download the group batches:

1. Connect to the SATCTLDB database on the satellite control server.
2. Issue the following SQL statement, where *satellite_id* is the satellite ID recorded in the SATADMIN.SATELLITES table:

```
UPDATE SATADMIN.SATELLITES SET sync_state='N' WHERE id='satellite_id'
```

Satellite Is in the Failed State at the Satellite Control Server

If a synchronization session is initiated and the SQLCODE -3935W is returned at the satellite, the satellite is in the failed state. That is, the satellite reported an error to the satellite control server during a previous synchronization session. When a satellite reports an error, it is automatically disabled from executing group batches. In this situation, you could create a fix batch to correct the problem on the satellite.

Related concepts:

- “Fix Batches” on page 219

Related tasks:

- “Identifying and Fixing a Failed Satellite” on page 337

Satellite Is Not Enabled at the Satellite Control Server

When a satellite is first created, it is disabled from executing its group batches. The disablement allows, for example, the staging of the deployment of the group satellites.

Before a satellite can execute the group batches for its particular application version, it must be enabled to execute the group batches. If a synchronization session is initiated and the satellite is not enabled to execute its group batches, the SQLCODE -3934W is issued.

The satellite may be disabled for the following reasons:

- The satellite is in the failed state. You must fix the satellite before enabling it to synchronize. See the topic on identifying and fixing a failed satellite for more information.
- The satellite is disabled for administrative reasons.

- The satellite is newly created.

If the SQLCODE -3934W warning is issued and you want this satellite to begin synchronizing, use the Satellite Administration Center to enable the satellite at the satellite control server.

Related tasks:

- “Identifying and Fixing a Failed Satellite” on page 337
- “Enabling a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Satellite’s Application Version Does Not Exist at the Satellite Control Server

When a satellite synchronizes, it passes its application version to the satellite control server. The satellite control server uses this information to determine which group batches and batch steps the satellite is to execute. If the satellite control server cannot find this application version for the satellite’s group in the satellite control database, the SQLCODE -3932W is returned. The synchronization session ends.

If this warning occurs, one of the following is possible:

- The application version set on the satellite is not correct.
You can use the **db2sync -s application_version** command or the db2SetSyncSession API to set the application version on the satellite. To display the current value of the application version on the satellite, use the **db2sync -g** command.
- The application version for the satellite’s group is not yet created at the satellite control server.

In this situation, the satellite may have been modified to use the next version of the application, but the application version is not yet created for the satellite’s group in the satellite control database. Use the Create Application Version window to create the application version for the group.

Note: The application version is case sensitive.

Related tasks:

- “Creating an application version : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Related reference:

- “db2sync - Start DB2 Synchronizer Command” in the *Command Reference*
- “db2SetSyncSession - Set Satellite Sync Session” in the *Administrative API Reference*

Application Version Is Not Set at the Satellite

When a satellite synchronizes, it passes its application version to the satellite control server. The satellite control server uses this information to determine which group batches and batch steps the satellite is to execute. If a synchronization session is initiated and the application version is not recorded locally on the satellite, the SQLCODE -3956N is returned. The synchronization session ends.

You can use the **db2sync -s *application_version*** command or the `db2SetSyncSession` API to set the application version on the satellite.

Note: The application version is case sensitive.

Related reference:

- “db2sync - Start DB2 Synchronizer Command” in the *Command Reference*
- “db2SetSyncSession - Set Satellite Sync Session” in the *Administrative API Reference*

Authentication Error Occurs at Satellite

If a satellite has not synchronized for some time and the password to the satellite control server changed before the satellite had a chance to download the changes during a synchronization session, the SQLCODE -1403 may be issued when the synchronization session is initiated. This SQLCODE indicates an authentication error. In this situation, the satellite does not have the correct password for the satellite control server in its `instance_path\security\satadmin.aut` file. See the topic on recreating or updating the `satadmin.aut` file for information on how to update this file.

An authentication error can also occur if the satellite ID is changed after the satellite has the correct authentication information in its `satadmin.aut` file. In this situation, the SQLCODE -3966 is issued with a reason code of 1. To recover from this error, you can:

- Use the **db2set** command to set the value of the DB2SATELLITEID registry variable back to its previous value.
- Use the **db2sync -t** command to start the DB2® Synchronizer application in test mode on the satellite. In this situation, the Connect to Control Database window opens, which you use to specify the authentication credentials to use to connect to the satellite control database. For information about using the Connect to Control Database window, refer to the online help that is available from the window.

Note: For the **db2sync -t** solution to work, you must have already created a satellite using the Satellite Administration Center, and the ID of the

new satellite must match the new value specified for the DB2SATELLITEID registry variable on the satellite.

Related tasks:

- “Recreating or Updating the satadmin.aut File on a Satellite” on page 346
- “Creating a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Related reference:

- “db2set - DB2 Profile Registry Command” in the *Command Reference*
- “db2sync - Start DB2 Synchronizer Command” in the *Command Reference*

Rebinding the db2satcs Stored Procedure Because of SQL0818

If a satellite receives an SQL0818 when synchronizing, the version of the db2satcs stored procedure does not match the version of the package in the satellite control database catalogs. Either the executable or the package is out of date.

Procedure:

To rebind the db2satcs stored procedure to the SATCTLDB database:

1. Open a command window.
2. Issue the following commands in this sequence:

```
db2set db2instance=db2_control_server_instance
db2 connect to satctldb
db2 bind db2installpath\bnd\db2satcs.bnd
```

Related reference:

- “CONNECT (Type 1) statement” in the *SQL Reference, Volume 2*
- “CONNECT (Type 2) statement” in the *SQL Reference, Volume 2*
- “BIND Command” in the *Command Reference*
- “db2set - DB2 Profile Registry Command” in the *Command Reference*

Identifying and Fixing Failed Satellites

The sections that follow describe how to use the Satellite Administration Center to identify and fix problems on failed satellites. For information about how to use the windows and notebooks of the Satellite Administration Center, and for information about the different views and icons that are available, refer to the online help that is available from the Satellite Administration Center.

Identifying and Fixing a Failed Satellite

If an error occurs when a satellite is synchronizing, the satellite stops synchronizing, and it reports the error to the satellite control server. The satellite control server then puts the satellite into the failed state, which prevents the satellite from synchronizing. Before the satellite can synchronize to execute its group batches, the satellite must be fixed.

Procedure:

To identify and fix the failed satellite so that it can return to executing its group batches:

1. Identify the failed satellite.
2. Obtain information about the failure.
3. Assign a fix batch to the satellite.

If the fix batch does not produce the correct results on the satellite, or the satellite reports an error while executing the fix batch, you need to debug the fix batch.

4. Return the repaired satellite to production.

Related concepts:

- “Fix Batches” on page 219

Related tasks:

- “Identifying the Failed Satellite” on page 337
- “Obtaining Information About the Failure on the Satellite” on page 339
- “Assigning a Fix Batch to the Satellite” on page 340
- “Debugging the Fix Batch” on page 342
- “Returning the Repaired Satellite to Production” on page 343

Identifying the Failed Satellite

The Satellite Administration Center uses roll-up views, which provides easy access to a quick, high-level snapshot of a group, or of an application version. The roll-up views display information about any failures that occur. That is, if a satellite reports a failure when it uploads its status during a synchronization session, you do not need to have the satellite details open to find out that a failure occurred. Instead, the icons that are displayed in the object tree indicate if an error has occurred.

If you want to view basic information about the icons that are used in the Satellite Administration Center, use the Show/Hide Legend icon on the tool bar to open the Legend window.

Procedure:

The following example describes how to identify a production satellite that reports an error, and what the error is. To find the failed production satellite, and the error that it reported:

1. Expand the Groups folder in the object tree.

The folder for the group that contains the failed satellite will have a red "X" superimposed over it to indicate that at least one production satellite reported an error while executing the group batches of a specific application version.

2. Select the folder for the group that contains the failed satellite.

So that you can easily identify when a production satellite reports an error, the icons that represent both the Satellites folder and the Application Version folder associated with the group contain red within the folder.

Note: The Application Version folder associated with the group only has red within the folder when one or more production satellites report an error while executing a group batch. The Satellites icon associated with the group has red within the folder if any satellite, either test or production, reports an error.

3. Within the folder for the group that contains the failed satellite:

- a. Click the Application Version folder to determine which application version was being executed by the production satellite (or satellites) that reported an error. The application version details view opens in the contents pane of the Satellite Administration Center.

The icon representing the application version that was being executed by a production satellite that reported an error has a red "X" superimposed over it as follows:



- b. Click the Satellites folder to determine which production satellite (or satellites) reported an error. The satellite details view opens.

The icon that represents the production satellite that reported the error is as follows:



The X over the icon indicates that the production satellite has failed when executing group batches. The gradation on the sphere in the icon indicates that the satellite is disabled.

Note: If a test satellite reports an error, it is also disabled from executing group batches.

After you identify the failed satellite, you obtain information about the failure.

Related tasks:

- “Obtaining Information About the Failure on the Satellite” on page 339
- “Satellite Administration Center icons : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Viewing application version details : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Viewing satellite details : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Obtaining Information About the Failure on the Satellite

When you identify the failed satellite, the next step is to obtain information about the failure that occurred. To perform this task, you view the logs for the satellite.

Procedure:

To view the logs for the satellite:

1. Right click on the failed satellite.
2. Select **Show Logs** from the pop-up menu.
The Show Logs window opens. The logs in the window are organized by date in descending order.
3. Select the log that records the failure and right click. Because the logs are organized by date in descending order, typically, the log that records the failure is the first log in the list.
4. Select **View Details** from the pop-up menu.
The Log Details window opens. This window displays the full set of information for the log that you selected. The information includes the batch that was being executed, as well as the batch step and the script that did not execute successfully. The log information also indicates whether the error resulted in an external or an internal return code.

You can also view the logs for the failed satellite by starting from the Logs folder in the object tree. To perform this task:

1. Select the Logs folder.
The Log Details view opens in the contents pane. The logs in the view are organized by date in descending order. You can use the sort and filter facilities that are available from the Satellite Administration Center to

modify the details view to reveal failed satellites. For example, you can filter the view to display the logs for a specific satellite, or only failed satellites.

2. Select the log that records the failure and right click.
3. Select **View Details** from the pop-up menu.

The Log Details window opens. This window displays the full set of information for the log record that you selected. The information includes the batch that was being executed, as well as the batch step and the script that did not execute successfully. The log information also indicates whether the error resulted in an external or an internal return code.

After you examine the logs to determine the cause of the failure and create a fix batch to correct the problem, you can assign the fix batch to the satellite.

Related concepts:

- “Internal and External Error Return Codes for Batch Steps” on page 345

Related tasks:

- “Assigning a Fix Batch to the Satellite” on page 340
- “Viewing details of a log entry : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Viewing log details : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Assigning a Fix Batch to the Satellite

When you have determined the problem that caused the satellite to fail, the next step is to attempt to fix the problem. If the satellite reported an error that occurred on other satellites, you may already have a fix batch to fix that error. Otherwise, use the information from the log to create a fix batch.

Note: You do not always need to use a fix batch to fix a satellite. For example, if a setup problem has occurred, you can enable the satellite (which will automatically be disabled when it fails), correct the environment problem, then have the satellite synchronize. You should always check the logs associated with the satellite before attempting to fix the satellite.

Procedure:

To assign a fix batch to the satellite:

1. Open the Satellite Details view.
2. Select the failed satellite in the Satellite Details view and right click.
3. Select **Fix** from the pop-up menu. The Fix Satellite window opens.

Use the Fix Satellite window to assign the fix batch that you want the satellite to execute, and the batch step where you want the satellite to begin executing the batch. You can use the ... push button to display the list of fix batches and unassigned batches that are available. If you do not have a batch that is suitable to fix the problem, create an unassigned batch and use it.

Note: The satellite will only be able to execute fix batches. Because the satellite is in fix mode, it cannot execute group batches. If the user attempts to synchronize the failed satellite before you enable it to execute the fix batch, the SQLCODE -3934W is returned at the satellite.

4. Click **OK**.
5. Select the failed satellite in the Satellite Details view and right click.
6. Select **Enable** from the pop-up menu.

When a satellite reports an error, its state changes to Failed in the Satellite Details view. In addition, the satellite is disabled. That is, the satellite cannot execute batches. The satellite must be enabled to execute the fix batch.

7. Click **OK**.
8. Have the user synchronize.
9. View the results of the execution of the fix batch.

To perform this task, you should view the logs for the satellite to determine whether the satellite executed the fix batch successfully. In addition, you can query the results of the fix batch. To perform the query, you can add a batch step to the fix batch that the satellite executed and have the satellite execute only that batch step, or you can use a different fix batch. If you are satisfied with the results of the fix, you are ready to promote the satellite back to executing its group batches. If you are satisfied with the results of the fix batch, you can return the repaired satellite to production. If the satellite reported an error while executing the fix batch, or you are not satisfied with the results of the fix batch, you must debug the fix batch.

Related tasks:

- “Returning the Repaired Satellite to Production” on page 343
- “Debugging the Fix Batch” on page 342
- “Creating a batch : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Viewing satellite details : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Debugging the Fix Batch

If the satellite reported an error when it executed the fix batch, or the results of the fix batch are not satisfactory, you need to debug the fix batch.

Procedure:

To debug the fix batch:

1. Determine the problem with the fix batch.

To perform this task, examine the logs for the satellite that executed the fix batch. If the log shows that an error occurred, you can begin the process of debugging the fix batch based on the error. If, however, an error did not occur, you can have the satellite execute a fix batch that queries the state of the satellite. You may have to try different queries to determine the problem.

2. Edit the fix batch to make the changes that you require.
3. Open the satellite details view.
4. Select the satellite that you want and right click.
5. Select **Edit** from the pop-up menu.

The Edit Satellite notebook opens.

6. On the Batches page, specify the fix batch that you want the satellite to execute, and the batch step where you want the satellite to begin executing the batch.
7. Enable the satellite to execute the fix batch, if necessary.

This step is only required if the satellite reported an error when it executed the fix batch. When a satellite reports an error, that satellite is automatically disabled from executing batches. If the satellite successfully executed the fix batch, but the results of the fix batch are not satisfactory, the satellite remains enabled to execute fix batches. You can check the satellite details view in the Satellite Administration Center to determine whether the satellite is enabled or disabled.

8. Have the satellite synchronize.
9. View the results of the execution of the fix batch.

To perform this task, you should view the logs for the satellite to determine whether the satellite executed the fix batch successfully. In addition, you can have the satellite execute another fix batch to query the results of the previous fix batch. If you are satisfied with the results of the fix, you are ready to promote the satellite back to executing its group batches. If you are not satisfied with the results of the fix, return to step 2 and repeat the procedure.

Related tasks:

- “Obtaining Information About the Failure on the Satellite” on page 339
- “Returning the Repaired Satellite to Production” on page 343
- “Editing a batch : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Enabling a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Promoting a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Viewing satellite details : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Returning the Repaired Satellite to Production

When the fix that you apply to the satellite produces the results that you want, the satellite can return to production. That is, the satellite can return to executing its group batches when it synchronizes.

Restrictions:

You should not return the satellite to production until you are fully satisfied with the results of the fix batch.

Procedure:

To return the satellite to production:

1. Open the satellite details view.
2. Select the satellite that you fixed and right click.
3. Select **Promote** from the pop-up menu.
The Promote Satellite window opens.
4. Depending on the fix that you applied, you may need to specify that the satellite resumes execution of one or more of its group batches at a different batch step than the next one to be executed. The fields of the Promote Satellite window indicate which group batches the satellite executes, and the batch step at which the satellite begins executing each group batch. Use the ... push button to specify the batch step where the satellite is to begin executing its group setup, update, or cleanup batch, as required.
5. Click **OK**.

If the satellite is already enabled (that is, it did not report an error when executing the fix batch), the next time that the satellite synchronizes, it will download and execute its group batches, beginning execution at the batch steps that you specified. If the satellite reported an error when it executed

the fix batch and the error is not important, you must enable the satellite before it can execute its group batches.

6. If the satellite is disabled, select it from the satellite details view and right click.
7. Select **Enable** from the pop-up menu.
8. Click **OK** when the Enable Satellite window opens to confirm that you want to enable this satellite to execute its group batches.

Related tasks:

- “Promoting a satellite : Satellite Administration Center help” in the *Help: Satellite Administration Center*
- “Viewing satellite details : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Running the DB2 Trace Facility on a Satellite

If requested to by DB2 Customer Service, you can use the DB2 trace facility (**db2trc**) on a satellite.

Restrictions:

You cannot issue this command from a remote DB2 CLP window.

Procedure:

To run the **db2trc** command against a satellite, include the command in a fix batch, or use a product that enables you to enter commands from a remote console and keyboard.

Because the formatted or flowed output files are typically much larger than the dump file, it is recommended that you issue the **db2trc** command with the **dmp** option. When the trace is dumped to a file, transmit the file to another DB2 Universal Database system. Then you can format the file before sending it to DB2 Customer Service.

Related reference:

- “db2trc - Trace Command” in the *Command Reference*

Satellite Software Version

The software version of a satellite is based on the release number used by DB2 Universal Database. You can find the official description of the release number in the C header file `db2ApiDf.h`. When a satellite synchronizes, it uploads its software version to the satellite control server.

You can find the software version of the satellite in the satellite details view.

You can use release number for diagnostic purposes. In this situation, different release numbers may have different fixes to correct known problems. You can also use the release number to stage upgrades of DB2® on the satellites.

Related tasks:

- “Viewing satellite details : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Internal and External Error Return Codes for Batch Steps

The error return codes are associated with the execution of a batch step by a satellite. The return code can be either an external return code or an internal return code. You can determine the type of the return code by viewing the log details in the Satellite Administration Center.

If the event associated with a log record involves an internal code failure, the error return code is displayed under the **Internal code** column. For example, assume that a satellite, when attempting to execute a script, encounters a file system full condition. This error code will be displayed under the **Internal code** column when the satellite returns the results of the synchronization session to the satellite control server.

If the script of a batch step does not execute successfully (according to its associated success code set), the error return code is displayed under the **External code column**.

Related tasks:

- “Viewing log details : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Satellite Progress File

Each satellite has its own log file to record information about its success or failure in executing the batch steps of the different batches that it executes during a synchronization session. The log file is called `progress.log`, and is located in the `instance_path\satellite` directory. The log file contains detailed information about the events that occurred on the satellite.

Recreating or Updating the `satadmin.aut` File on a Satellite

If the `satadmin.aut` file on a satellite is either destroyed or out-of-date, you can have the file either recreated or updated on the satellite.

Prerequisites:

Before you can perform this task, you must either set the `DB2SATELLITEID` registry variable to the satellite ID, or you must be logged on as the user ID that will be used to synchronize.

Procedure:

To recreate or update the `satadmin.aut` file:

1. Start the DB2 Synchronizer application in test mode by issuing the following command on the satellite:

```
db2sync -t
```

2. Click the **Test** push button.

If the `satadmin.aut` file no longer exists on the satellite or is out-of-date, no authentication credentials exist that the satellite can use to authenticate with the satellite control server. In this situation, the `SQLCODE -3966` is issued, and the **Connect to Control Database** window opens. Use this window to specify the authentication credentials to use to connect to the satellite control database. For information about using the **Connect to Control Database** window, refer to the online help that is available from the window. When you provide the correct user ID and password, the satellite control server will authenticate the satellite, and the `satadmin.aut` file will be either recreated or updated on the satellite.

3. Close the DB2 Synchronizer window.
4. Start another synchronization session via the application that you use to synchronize.

Related concepts:

- “Authentication Credentials” on page 221
- “Storage of Authentication Credentials on Satellites” on page 223

Related tasks:

- “Setting the DB2SATELLITEID Registry Variable on a Satellite” on page 250

Related reference:

- “db2sync - Start DB2 Synchronizer Command” in the *Command Reference*

Determining Synchronization Errors when Log Details Are Truncated

The log details information that is available from the Satellite Administration Center about the synchronization session for a satellite can be a maximum of 256 KB for each log. If log truncation occurs, the last 256 KB of the log is retained. If you are using batches that result in more than 256 KB of detailed log information being written to the Satellite Administration Center and a satellite reports an error during a synchronization session (that is, the satellite is in the failed state), the log details may not indicate the problem.

Procedure:

If the error is not recorded in the log that is retained:

1. Issue a query against the SATELLITES table.

The SATELLITES table contains the ID column for the satellite, and the GRP_BEF_BATCH_STEP, the GRP_REP_BATCH_STEP, and the GRP_AFT_BATCH_STEP columns. These three columns indicate the last batch step that the satellite executed for the setup, update, and cleanup batches, respectively. For example, if you issue a query for the satellite and obtain a value from the GRP_BEF_BATCH_STEP column of 3, and your setup batch has seven batch steps, you know that the satellite encountered an error at batch step 3.

2. Determine the error code that was returned by the failed batch step.

To determine the error, open the satellite details view. From the satellite details view, select the failed satellite and view its logs. The Show Logs window opens, which displays the internal and external return codes.

Related concepts:

- “Internal and External Error Return Codes for Batch Steps” on page 345

Related tasks:

- “Viewing satellite details : Satellite Administration Center help” in the *Help: Satellite Administration Center*

Related reference:

- Appendix C, “General Administration Tables” on page 357

Part 3. Appendixes

Appendix A. Unique Characteristics of DB2 Satellite Edition Satellites

Before you migrate a DB2 Satellite Edition satellite to DB2 Universal Database, Version 8, you should review the following list of functions and features that are either not supported on DB2 Satellite Edition, or are unique to DB2 Satellite Edition. Reviewing this list should help you ensure that your applications will continue to operate.

DB2 Satellite Edition did not provide the following:

- Communications using APPC, IPX/SPX, NetBios, and NamedPipes. Only TCP/IP is supported.
- DRDA Application Server functionality for DRDA Application Requestors.
- A DB2 Administration Server (DAS). If you want to use the Control Center to administer a DB2 Satellite Edition satellite, not all of the Control Center functionality is available.
 - Satellites cannot be discovered using the DB2 discovery function. To connect to a satellite, you must create the node and database directory entries for that satellite using the Control Center, the Configuration Assistant, or manually, using the CLP.
 - You cannot schedule scripts for execution on a satellite.
 - You cannot use the browse function of a remote Control Center to view the satellite's directory structure or files.
 - You cannot grant or revoke privileges on a satellite from the Control Center.
 - You cannot use the full function of some SmartGuides.
- The DB2 graphical user interface tools, including:
 - The Control Center
 - The Configuration Assistant
 - The Event Analyzer
 - The Command Center
 - The Information Center.

Note: If you require these tools on a DB2 Satellite Edition satellite, you can install the DB2 UDB Administrative Client on the satellite.

- The ability to create the SAMPLE database.
- The DB2 Governor.
- The coding samples that are typically used by database administrators or systems staff.

- Lightweight Directory Access Protocol (LDAP).
- Distributed Computing Environment (DCE).
- Simple Network Management Protocol (SNMP).
- DB2 documentation in HTML, PostScript, or online help.
- Remote monitoring.

Some default configuration parameter values are different for DB2 Satellite Edition satellites than for other DB2 Universal Database instances and databases. Configuration parameters that apply to satellites are identified by Satellite database server with local clients in the table at the beginning of each relevant parameter. See the DB2 Universal Database Version 6 information for details.

Appendix B. Overview of the Satellite Control Tables

Two types of tables exist in the satellite control database: general administration and replication administration. These tables all exist under the SATADMIN schema.

General administration

The general administration tables are those that are required to administer groups of satellites, without any provisions for the administration of data replication. That is, these tables are part of the infrastructure required to support the administration of groups of similar satellites. These tables contain the metadata associated with synchronization (such as configuration information), but do not contain the metadata associated with data synchronization.

Replication administration

The replication administration tables simplify the administration of data synchronization for groups of similar satellites. These tables contain group-level replication characteristics that apply to each satellite, *unless* you have specified satellite-specific customizations that override the group-level replication characteristics. For example, one replication characteristic is the specific rows of tables to be replicated. This does not have to be the same for all the group satellites.

The following table provides an overview of the type of information that is contained in the general administration and replication administration tables.

Table 14. Overview of Satellite Control Tables

Table Name	Description
General Administration	
SATELLITES	Contains static and dynamic information that describes the satellites in the satellite environment.
GROUPS	Contains information about the groups to which the satellites belong.
SCRIPTS	Contains information about the operating system commands, DB2 [®] commands, and SQL statements that satellites execute when they synchronize.

Table 14. Overview of Satellite Control Tables (continued)

Table Name	Description
BATCHES	Contains information about the batches, which are named collections of ordered batch steps.
BATCH_STEPS	Contains information about each batch step. Each batch step is the combination of a script and a success code set. In addition, if the script executes against either a DB2 instance or database, the batch step also includes an execution target and an authentication credential.
GROUP_BATCHES	Contains information about the group batches that are associated with an application version for a specific group.
APP_VERSIONS	Contains information about the application versions that are defined at the satellite control server.
TARGETS	Contains information about the DB2 instances and databases against which scripts are executed when a satellite synchronizes.
TARGET_AUTH	Contains information about the authentication credentials (user ID and password) that are required for authentication against execution targets.
SUCCESS_CODES	Contains descriptive information about the named sets of return-code relations that are stored in the SUCCESS_RELATIONS table.
SUCCESS_RELATIONS	Contains information about the return-code relations, the conjunction of which defines the range of acceptable return codes for each operating system or DB2 command, or SQL statement that is executed by a satellite during a synchronization session.
LOG	Contains information about the satellite synchronization sessions.
Replication Administration	
GROUP_SUBSCR_SETS	Contains information about the replication subscription sets that are associated with the groups.
GRP_HOR_DATASLICES	Contains information about the predicates that define the rows that each satellite replicates from each source-target table pair for each group-level subscription set.

Table 14. Overview of Satellite Control Tables (continued)

Table Name	Description
GROUP_APPVER_PARMS	Contains information about the default replication control servers that are used by all satellites in a group.
SAT_HOR_DATASLICES	Contains information about the predicates that define the rows that each satellite replicates from each source-target target pair for each group-level subscription set. If a specific entry is not defined for a satellite, the satellite replicates data based on the information in the GRP_HOR_DATASLICES table
SAT_APPVER_PARMS	Contains information about the replication control servers that are used by a satellite, if that satellite does not use its group's replication control servers (as recorded in the GROUP_APPVER_PARMS table).

Related reference:

- Appendix C, "General Administration Tables" on page 357

Appendix C. General Administration Tables

Following is a detailed description of the general administration tables of the satellite control database:

- "SATELLITES table"
- "GROUPS table" on page 362
- "SCRIPTS table" on page 363
- "BATCHES table" on page 364
- "BATCH_STEPS table" on page 365
- "GROUP_BATCHES table" on page 367
- "APP_VERSIONS table" on page 368
- "TARGETS table" on page 368
- "TARGET_AUTH table" on page 369
- "SUCCESS_CODES table" on page 370
- "SUCCESS_RELATIONS table" on page 370
- "LOG table" on page 371

SATELLITES table:

The SATELLITES table contains information that describes the satellites in the satellite environment. The structure of the table is as follows:

SATELLITES Table			
Column	Data Type	Constraints	Description
ID	VARCHAR(20)	Primary key.	The unique identifier for the satellite. The ID must be unique in the satellite control server. The ID is equal to the Apply qualifier that is used for data replication, and is usually the logon ID of the satellite user.
GROUP	INTEGER	Foreign key: GROUPS:ID.	The group to which the satellite belongs.
FIRST_NAME	VARCHAR(18)		The first name of the satellite user.
LAST_NAME	VARCHAR(18)		The last name of the satellite user.

SATELLITES Table			
Column	Data Type	Constraints	Description
SUB_GROUP	VARCHAR(18)		The subgroup that the satellite is assigned to. This information can be used to stage the deployment of the group.
DB2_VERSION	BIGINT		The version of DB2 that is on the satellite.
APP_VERSION	VARCHAR(18)		The version of the end-user application that is on the satellite.
ONLINE	CHAR(1)	NOT NULL; values are Y or N. Default is N.	<p>Y The satellite is currently connected to the satellite control database for synchronization. Either the satellite is downloading its batches for execution, or reporting the results of the batch execution.</p> <p>N The satellite is not connected to the satellite control database.</p>
ENABLED	CHAR(1)	NOT NULL; values are Y or N. Default is N.	<p>Y The satellite is enabled to execute either group-level or fix batches.</p> <p>N The satellite is not enabled to execute batches.</p>

SATELLITES Table

Column	Data Type	Constraints	Description
TYPE	CHAR(1)	NOT NULL; values are P or T. Default is P.	<p>P A production satellite; the satellite executes only production batch steps.</p> <p>T A test satellite; the satellite executes only test batch steps.</p> <p>Also see BATCH_STEPS:TYPE and BATCHES:TYPE.</p>
FAILED	CHAR(1)	NOT NULL; values are Y or N. Default is N.	<p>Y An error occurred during the last synchronization session (for example, a script failed).</p> <p>N The satellite did not report an error during the last synchronization session.</p>
TIME_OF_FAILURE	TIMESTAMP		The time at which the value for SATELLITES:FAILED was set to Y.

SATELLITES Table			
Column	Data Type	Constraints	Description
SYNC_STATE	CHAR(1)	NOT NULL; values are N, B, or R. Default is N.	<p>The state of the satellite with respect to synchronization. The possible values are:</p> <p>N The satellite has not yet synchronized.</p> <p>B The satellite has downloaded its batches for execution.</p> <p>R The satellite has uploaded the results of the batch execution.</p>
LAST_STATE_CHANGE	TIMESTAMP		The time the value of SATELLITES:SYNC_STATE was last updated.
BATCH_MODE	CHAR(1)	NOT NULL; values are F or G. Default is G.	<p>F The satellite will execute its fix batch when it next synchronizes.</p> <p>G The satellite will execute its group batches when it next synchronizes.</p>

SATELLITES Table

Column	Data Type	Constraints	Description
PASSWORD_CHANGED	CHAR(1)	NOT NULL; values are Y or N. Default is N.	<p>Y If the group-level authentication credential changes, refresh the satellite's local password file with the changed password the next time that the satellite synchronizes.</p> <p>N The satellite's local password file does not need to be refreshed. The group-level authentication credential is not changed.</p>
GRP_BEF_BATCH_STEP	INTEGER	NOT NULL; default is 1.	The batch step of the group setup batch that the satellite will next execute. The satellite will execute, in sequence, all batch steps of the setup batch that have a value either equal to, or greater than, the value of this column.
GRP-REP_BATCH_STEP	INTEGER	NOT NULL; default is 1.	The batch step of the group update batch that the satellite will next execute. The satellite will execute, in sequence, all batch steps of the update batch that have a value either equal to, or greater than, the value of this column.

SATELLITES Table			
Column	Data Type	Constraints	Description
GRP_AFT_BATCH_STEP	INTEGER	NOT NULL; default is 1.	The batch step of the group cleanup batch that the satellite will next execute. The satellite will execute, in sequence, all batch steps of the cleanup batch that have a value either equal to, or greater than, the value of this column.
FIX_BATCH	VARCHAR(18)	Can be NULL; when set, must match value in BATCHES:ID.	When SATELLITES:BATCH_MODE is set to F, the satellite will execute the batch indicated in this column.
FIX_BATCH_STEP	INTEGER	NOT NULL; default is 1.	The batch step of the fix batch that the satellite will next execute. The satellite will execute, in sequence, all batch steps of the fix batch that have a value either equal to, or greater than, the value of this column.

GROUPS table:

The GROUPS table contains information about the groups that are in the satellite environment. The structure of the table is as follows:

GROUPS			
Column	Data Type	Constraints	Description
ID	INTEGER	Primary key.	A unique, internal ID.
NAME	VARCHAR(18)	NOT NULL; must be unique.	The unique group identifier. This value is not a primary key, which enables the group to be renamed.

GROUPS			
Column	Data Type	Constraints	Description
REMOTE_CS_AUTH	INTEGER	Foreign key: TARGET_AUTH:ID.	The group-level authentication credential (user ID and password). Each satellite in the group uses this authentication credential when it connects to the SATCTLDB database for synchronization. The user ID of the authentication credential possesses EXECUTE authority on the stored procedures that can access and update the satellite control tables.
DESCRIPTION	VARCHAR(128)		A description of the group.

SCRIPTS table:

The SCRIPTS table contains information about the script component of a batch step. The structure of the table is as follows:

SCRIPTS			
Column	Data Type	Constraints	Description
ID	INTEGER	Primary key.	An internal, unique identifier for the script. The ID is not the file name.
SCRIPT	CLOB(512KB)	NOT_NULL	The script contents.
FILENAME	VARCHAR(255)	NOT_NULL; unique.	The file name of the script. This is the name under which the script is saved on the satellite. Operating system scripts must have an extension of either .bat or .cmd. The file name is not a primary key, so that it can be modified.

SCRIPTS			
Column	Data Type	Constraints	Description
PARAMETERIZED	CHAR(1)	NOT NULL; values are Y or N. Default is N.	Y The script is parameterized. N The script is not parameterized.
TYPE	CHAR(1)	NOT NULL; values are I, D, or O.	I The script executes against a DB2 instance. D The script executes against a database. O The script executes against the operating system.
STATEMNT_TERM_CHAR	CHAR(1)	NOT NULL; default is semicolon (;).	The termination character for scripts that execute against a DB2 instance or database.
DESCRIPTION	VARCHAR(128)		A description of the script.

BATCHES table:

The BATCHES table contains information about all the batches that are available. The structure of the table is as follows:

BATCHES			
Column	Data Type	Constraints	Description
ID	VARCHAR(18)	Primary key, the value must be greater than, or equal to, 0.	The name of the batch.

BATCHES			
Column	Data Type	Constraints	Description
MOD_LEVEL	INTEGER	Primary key.	The modification level of the batch. The value is 0 if the batch is either an unassigned or a fix batch (that is, MODE is either U or F).
TYPE	CHAR(1)	NOT NULL; values are T, P, O, or N. Default is N.	<p>T A test-level group batch.</p> <p>P A production-level group batch.</p> <p>O An obsolete-level group batch.</p> <p>N Not applicable. The batch is either an unassigned batch, or a fix batch.</p>
MODE	CHAR(1)	NOT NULL; values are F, G, or U. Default is U.	<p>F A fix batch.</p> <p>G A group batch.</p> <p>U An unassigned batch.</p>
GROUP	INTEGER	Can be NULL. If set (that is, MODE is set to G), the value must match GROUPS:ID.	If the batch is a group batch, it must be associated with a particular group.
DESCRIPTION	VARCHAR(128)		A description of the batch.

BATCH_STEPS table:

The BATCH_STEPS table contains information about every batch step, including the script, execution target, authentication credential, and success code set. The structure of the table is as follows:

BATCH_STEPS			
Column	Data Type	Constraints	Description
BATCH	VARCHAR(18)	Primary key; foreign key: BATCHES.ID.	The name of the batch to which the batch step is assigned.
MOD_LEVEL	INTEGER	Primary key; foreign key BATCHES:MOD_LEVEL. The default is 0. The value must be greater than, or equal to, 0.	The modification level of the batch step.
STEP	INTEGER	Primary key; the value must be greater than 0. The order is sequential, and no gaps can exist in the numbering sequence.	Batch steps are scripts that are ordered for sequential execution.
SCRIPT	INTEGER	Foreign key: SCRIPTS.ID.	The script that is executed for this batch step.
TYPE	CHAR(1)	NOT NULL; values are T, P, O, or N. Default is T.	<p>T A batch step in a test-level group batch.</p> <p>P A batch step in a production-level group batch.</p> <p>O A batch step in an obsolete-level group batch.</p> <p>N Not applicable. The batch step is in either an unassigned batch, or a fix batch.</p>
SUCCESS_CODES	VARCHAR(18)	Foreign key: SUCCESS_CODES.ID.	The named success code set that defines whether the script executes successfully.

BATCH_STEPS			
Column	Data Type	Constraints	Description
EXEC_TARGET_ALIAS	VARCHAR(255)	Foreign key: TARGETS:ALIAS.	The alias of the DB2 instance or database against which the script executes.
EXEC_TARGET_TYPE	CHAR(1)	Foreign key: TARGETS:TYPE; values are I, D, or O.	<p>I The script executes against a DB2 instance.</p> <p>D The script executes against a DB2 database.</p> <p>O The script executes against the operating system.</p>

GROUP_BATCHES table:

The GROUP_BATCHES table contains information about the batches that are associated with a group's application version. The group satellites execute these batches when they synchronize. The structure of the table is as follows:

GROUP_BATCHES			
Column	Data Type	Constraints	Description
GROUP	INTEGER	Primary key; foreign key: GROUPS:ID.	The group to which the group batch is assigned.
VERSION	VARCHAR(18)	Primary key; foreign key: APP_VERSIONS:APP_VERSION.	The application version with which the batch is associated. Only the satellites that belong to the specified group and are at the same application version can execute the batches.
PROD_MOD_LEVEL	INTEGER	NOT NULL; default is -1. The value must be greater than, or equal to, -1.	The modification level of the setup, update, and cleanup batches that are in production for the group.

GROUP_BATCHES			
Column	Data Type	Constraints	Description
TEST_MOD_LEVEL	INTEGER	NOT NULL; default is 0. The value must be greater than, or equal to, -1.	The modification level of the setup, update, and cleanup batches that are in test for the group.
BEFORE_BATCH	VARCHAR(18)	NULL	The setup batch for the group and application version.
REPEAT_BATCH	VARCHAR(18)	NULL	The update batch for the group and application version.
AFTER_BATCH	VARCHAR(18)	NULL	The cleanup batch for the group and application version.
DESCRIPTION	VARCHAR(128)		A description of this version of the group batches.

APP_VERSIONS table:

The APP_VERSIONS table contains information about the application versions that are available in the satellite environment. The structure of the table is as follows:

APP_VERSIONS			
Column	Data Type	Constraints	Description
APP_VERSION	VARCHAR(18)	NOT NULL; primary key	The name of the application version.
DESCRIPTION	VARCHAR(128)		A description of the application version.

TARGETS table:

The TARGETS table contains information about the execution targets. The structure of the table is as follows:

TARGETS			
Column	Data Type	Constraints	Description
ALIAS	VARCHAR(255)	Primary key	The alias of the DB2 instance or database. The alias is used for an ATTACH or a CONNECT.
TYPE	CHAR(1)	Primary key; values are I, D, or O.	<p>I The execution target is a DB2 instance.</p> <p>D The execution target is a DB2 database.</p> <p>O The execution target is the operating system on the satellite.</p>
AUTH_INFO	INTEGER	NOT NULL; foreign key: TARGET_AUTH:ID.	The name of the authentication credential that is required to authenticate against either the DB2 instance or the database. An authentication credential is not required if the execution target is the operating system.

TARGET_AUTH table:

The TARGET_AUTH table contains information about the authentication credential (user ID and password pair) that each script requires for authentication against either a DB2 instance or database. The structure of the table is as follows:

TARGET_AUTH			
Column	Data Type	Constraints	Description
ID	INTEGER	Primary key	An internal, unique identifier for the authentication credential. The ID is not the name.

TARGET_AUTH			
Column	Data Type	Constraints	Description
NAME	VARCHAR(18)	NOT NULL; the name must be unique.	The unique name of the authentication credential.
USERID	VARCHAR(256)	NOT_NULL	The authentication ID.
PASSWORD_ENCRYPTED	VARCHAR(256)	NOT_NULL; FOR BIT DATA	The encrypted password.

SUCCESS_CODES table:

The SUCCESS_CODES table contains information about the named sets of success codes. The structure of the table is as follows:

SUCCESS_CODES			
Column	Data Type	Constraints	Description
ID	VARCHAR(18)	Primary key	The unique name for the set of success code relations.
DESCRIPTION	VARCHAR(128)		A description of the success code set.

SUCCESS_RELATIONS table:

The SUCCESS_RELATIONS table contains information about the relations of the codes in the success code sets. The structure of the table is as follows:

SUCCESS_RELATIONS			
Column	Data Type	Constraints	Description
ID	VARCHAR(18)	Foreign key: SUCCESS_CODES:ID.	The name of the success code set to which this relation belongs.
RELATION	CHAR(1)	NOT_NULL; values are <, =, and >.	A script executes successfully if its return code is within the parameters defined by RELATION and CODE.
CODE	INTEGER	NOT_NULL	A return code for an operating system script, an SQLCODE for an SQL statement or a DB2 command script.

LOG table:

The LOG table contains records about synchronization events. The structure of the table is as follows:

LOG			
Column	Data Type	Constraints	Description
EVENT_TIME	TIMESTAMP	Primary key; NOT_NULL	The time at which the event occurred.
DUP_TIME_SEQ_NO	INTEGER	Primary key; NOT NULL; default 0.	To accommodate events that occur simultaneously.
TYPE	VARCHAR(18)	NOT_NULL	The type of the event: RETRIEVED_BATCHES The satellite downloaded its batches from the satellite control server. LOGD_BATCH_RESULTS The satellite reported the results of its synchronization session.
STATE	CHAR(1)	NOT_NULL; values are S, F, N, or U.	S The event succeeded. F The event failed. N The state is not applicable to the event, or no information is available. U The state is unknown.
INTERNAL_RC	INTEGER	NOT_NULL; default 0	If the associated event was generated by the synchronizing system, the return code is stored here.

LOG			
Column	Data Type	Constraints	Description
EXTERNAL_RC	INTEGER	NOT_NULL; default 0	If the associated event was generated by scripts or programs executed by the synchronization system, the return code is stored here. For example, if a script fails during a synchronization session, the return code is stored here.
DETAILS	CLOB(256 KB)		Information describing the event.
ASSOC_OBJECT	VARCHAR(20)		The unique ID of the object associated with the event (for example, a satellite).
ASSOC_OBJECT_TYPE	CHAR(1)	Values are G or S.	The type of object referenced in LOG:ASSOC_OBJECT: G The object is a group. S The object is a satellite. A The object is an application version.

Appendix D. Naming rules

Unless otherwise specified, all names can include the following characters:

- A through Z. When used in most names, characters A through Z are converted from lowercase to uppercase.
- 0 through 9
- @, #, \$, and _ (underscore)

Names cannot begin with a number or with the underscore character.

Do not use SQL reserved words to name tables, views, columns, indexes, or authorization IDs.

There are other special characters that might work separately depending on your operating system and where you are working with DB2. However, while they might work, there is no guarantee that they will work. It is not recommended that you use these other special characters when naming objects in your database.

You also need to consider object naming rules, workstation naming rules, naming rules in an NLS environment, and naming rules in a Unicode environment.

Related concepts:

- “General rules for naming objects and users” in the *Administration Guide: Implementation*
- Appendix E, “DB2 object naming rules” on page 375
- Appendix F, “Workstation naming rules” on page 377
- Appendix G, “User, userID and group naming rules” on page 379
- “Federated database object naming rules” in the *Quick Beginnings for DB2 Servers*

Appendix E. DB2 object naming rules

All objects follow the General Naming Rules. In addition, some objects have additional restrictions shown below.

Table 15. Database, database alias and instance naming Rules

Objects	Guidelines
<ul style="list-style-type: none">• Databases• Database aliases• Instances	<ul style="list-style-type: none">• Database names must be unique within the location in which they are cataloged. On UNIX-based implementations of DB2, this location is a directory path, while on Windows[®] implementations, it is a logical disk.• Database alias names must be unique within the system database directory. When a new database is created, the alias defaults to the database name. As a result, you cannot create a database using a name that exists as a database alias, even if there is no database with that name.• Database, database alias and instance names can have up to 8 bytes.• On Windows NT, Windows 2000, Windows XP and Windows .NET systems, no instance can have the same name as a service name. <p>Note: To avoid potential problems, do not use the special characters @, #, and \$ in a database name if you intend to use the database in a communications environment. Also, because these characters are not common to all keyboards, do not use them if you plan to use the database in another language.</p>

Table 16. Database Object Naming Rules

Objects	Guidelines
<ul style="list-style-type: none"> • Aliases • Buffer pools • Columns • Event monitors • Indexes • Methods • Nodegroups • Packages • Package versions • Schemas • Stored procedures • Tables • Table spaces • Triggers • UDFs • UDTs • Views 	<p>Can contain up to 18 bytes <i>except</i> for the following:</p> <ul style="list-style-type: none"> • Table names (including view names, summary table names, alias names, and correlation names), which can contain up to 128 bytes • Package names, which can contain up to 8 bytes • Schema names, which can contain up to 30 bytes • Package versions, which can contain up to 64 bytes • Object names can also include: <ul style="list-style-type: none"> – valid accented characters (such as ö) – multibyte characters, except multibyte spaces (for multibyte environments) • Package names and package versions can also include periods (.), hyphens (-), and colons (:).

Related concepts:

- Appendix D, “Naming rules” on page 373

Appendix F. Workstation naming rules

A *workstation name* specifies the NetBIOS name for a database server, database client, or DB2® Personal Edition that resides on the local workstation. This name is stored in the database manager configuration file. The workstation name is known as the *workstation nname*.

In addition, the name you specify:

- Can contain 1 to 8 characters
- Cannot include &, #, or @
- Must be unique within the network

In a partitioned database system, there is still only one workstation *nname* that represents the entire partitioned database system, but each node has its own derived unique NetBIOS *nname*.

The workstation *nname* that represents the partitioned database system is stored in the database manager configuration file for the database partition server that owns the instance.

Each node's unique *nname* is a derived combination of the workstation *nname* and the node number.

If a node does not own an instance, its NetBIOS *nname* is derived as follows:

1. The first character of the instance-owning machine's workstation *nname* is used as the first character of the node's NetBIOS *nname*.
2. The next 1 to 3 characters represent the node number. The range is from 1 to 999.
3. The remaining characters are taken from the instance-owning machine's workstation *nname*. The number of remaining characters depends on the length of the instance-owning machine's workstation *nname*. This number can be from 0 to 4.

For example:

Instance-Owning Machine's Workstation <i>nname</i>	Node Number	Derived Node NetBIOS <i>nname</i>
GEORGE	3	G3ORGE
A	7	A7
B2	94	B942
N0076543	21	N216543

Instance-Ownning Machine's <i>Workstation nname</i>	Node Number	Derived Node NetBIOS <i>nname</i>
GEORGE5	1	G1RGE5

If you have changed the default workstation *nname* during the installation, the workstation *nname*'s last 4 characters should be unique across the NetBIOS network to minimize the chance of deriving a conflicting NetBIOS *nname*.

Related concepts:

- Appendix D, "Naming rules" on page 373

Appendix G. User, userID and group naming rules

Table 17. User, userID and group naming rules

Objects	Guidelines
<ul style="list-style-type: none">• Group names• User names• User IDs	<ul style="list-style-type: none">• Group names can contain up to 8 bytes.• User IDs on UNIX-based systems can contain up to 8 characters.• User names on Windows® can contain up to 30 characters. Windows NT, Windows 2000, Windows XP and Windows .NET currently have a practical limit of 20 characters.• When not Client authentication, non-Windows 32-bit clients connecting to Windows NT, Windows 2000, Windows XP and Windows .NET with user names longer than 8 characters are supported when the user name and password are specified explicitly.• Names and IDs cannot:<ul style="list-style-type: none">– Be USERS, ADMINS, GUESTS, PUBLIC, LOCAL or any SQL reserved word– Begin with IBM, SQL or SYS.– Include accented characters.

Notes:

1. Some operating systems allow case sensitive user IDs and passwords. You should check your operating system documentation to see if this is the case.
2. The authorization ID returned from a successful CONNECT or ATTACH is truncated to 8 characters. An ellipsis (...) is appended to the authorization ID and the SQLWARN fields contain warnings to indicate truncation.

Related concepts:

- Appendix D, "Naming rules" on page 373
- "Federated database object naming rules" in the *Quick Beginnings for DB2 Servers*

Appendix H. Changing the DB2 interface language (UNIX)

The interface language of DB2 is the language that appears in messages, help, and graphical tool interfaces. When installing DB2, you have the option of installing support for one or more languages. If, at some time after installation, you want to change the interface language for DB2 to one of the other installed interface languages, use the steps outlined in this task.

Do not confuse languages supported by DB2 with languages supported by the DB2 interface. Languages supported by DB2, that is, languages that *data* can exist in, are a superset of languages supported by the DB2 interface.

Prerequisites:

Support for the DB2 interface language you want to use must be installed on your system. DB2 interface language support is selected and installed when you install DB2 using the DB2 Setup wizard. If you change the interface language of DB2 to a supported interface language that has not been installed, the DB2 interface language will default to the operating system language first, and if that is not supported, English.

Procedure:

To change the DB2 interface language on UNIX systems, set the LANG environment variable to the desired locale.

For example, to interface with DB2 in French using DB2 for AIX, you must have French language support installed and you must set the LANG environment variable to a French locale, for example, fr_FR.

Related reference:

- “Supported territory codes and code pages” in the *Administration Guide: Planning*
- “Supported DB2 interface languages, locales, and code pages” in the *Quick Beginnings for DB2 Servers*

Appendix I. Changing the DB2 interface language (Windows)

The interface language of DB2 is the language that appears in messages, help, and graphical tool interfaces. When installing DB2, you have the option of installing support for one or more languages. If, at some time after installation, you want to change the interface language for DB2 to one of the other installed interface languages, use the steps outlined in this task.

Do not confuse languages supported by DB2 with languages supported by the DB2 interface. Languages supported by DB2, that is, languages that *data* can exist in, are a superset of languages supported by the DB2 interface.

Prerequisites:

The DB2 interface language you want to use must be installed on your system. DB2 interface languages are selected and installed when you install DB2 using the DB2 Setup wizard. If you change the interface language of DB2 to a supported interface language that has not been installed, the DB2 interface language will default to the operating system language first, and if that is not supported, English.

Procedure:

Changing the interface language for DB2 on Windows requires that you change the default language setting for your Windows operating system.

To change the DB2 interface language on Windows:

1. Through the Control Panel on your Windows operating system, select **Regional Options**.
2. In the Regional Options dialog window, change the default language setting for the system to the language in which you want to interface with DB2.

Refer to your operating system help for additional information about changing the default system language.

Related reference:

- “Supported territory codes and code pages” in the *Administration Guide: Planning*
- “Supported DB2 interface languages, locales, and code pages” in the *Quick Beginnings for DB2 Servers*

Appendix J. DB2 Universal Database technical information

Overview of DB2 Universal Database technical information

DB2 Universal Database technical information can be obtained in the following formats:

- Books (PDF and hard-copy formats)
- A topic tree (HTML format)
- Help for DB2 tools (HTML format)
- Sample programs (HTML format)
- Command line help
- Tutorials

This section is an overview of the technical information that is provided and how you can access it.

Categories of DB2 technical information

The DB2 technical information is categorized by the following headings:

- Core DB2 information
- Administration information
- Application development information
- Business intelligence information
- DB2 Connect information
- Getting started information
- Tutorial information
- Optional component information
- Release notes

The following tables describe, for each book in the DB2 library, the information needed to order the hard copy, print or view the PDF, or locate the HTML directory for that book. A full description of each of the books in the DB2 library is available from the IBM Publications Center at www.ibm.com/shop/publications/order

The installation directory for the HTML documentation CD differs for each category of information:

htmlcdpath/doc/htmlcd/%L/category

where:

- *htmlcdpath* is the directory where the HTML CD is installed.
- *%L* is the language identifier. For example, *en_US*.
- *category* is the category identifier. For example, *core* for the core DB2 information.

In the PDF file name column in the following tables, the character in the sixth position of the file name indicates the language version of a book. For example, the file name *db2d1e80* identifies the English version of the *Administration Guide: Planning* and the file name *db2d1g80* identifies the German version of the same book. The following letters are used in the sixth position of the file name to indicate the language version:

Language	Identifier
Arabic	w
Brazilian Portuguese	b
Bulgarian	u
Croatian	9
Czech	x
Danish	d
Dutch	q
English	e
Finnish	y
French	f
German	g
Greek	a
Hungarian	h
Italian	i
Japanese	j
Korean	k
Norwegian	n
Polish	p
Portuguese	v
Romanian	8
Russian	r
Simp. Chinese	c
Slovakian	7
Slovenian	l
Spanish	z
Swedish	s
Trad. Chinese	t
Turkish	m

No form number indicates that the book is only available online and does not have a printed version.

Core DB2 information

The information in this category cover DB2 topics that are fundamental to all DB2 users. You will find the information in this category useful whether you are a programmer, a database administrator, or you work with DB2 Connect, DB2 Warehouse Manager, or other DB2 products.

The installation directory for this category is `doc/htmlcd/%L/core`.

Table 18. Core DB2 information

Name	Form Number	PDF File Name
<i>IBM DB2 Universal Database Command Reference</i>	SC09-4828	db2n0x80
<i>IBM DB2 Universal Database Glossary</i>	No form number	db2t0x80
<i>IBM DB2 Universal Database Master Index</i>	SC09-4839	db2w0x80
<i>IBM DB2 Universal Database Message Reference, Volume 1</i>	GC09-4840	db2m1x80
<i>IBM DB2 Universal Database Message Reference, Volume 2</i>	GC09-4841	db2m2x80
<i>IBM DB2 Universal Database What's New</i>	SC09-4848	db2q0x80

Administration information

The information in this category covers those topics required to effectively design, implement, and maintain DB2 databases, data warehouses, and federated systems.

The installation directory for this category is `doc/htmlcd/%L/admin`.

Table 19. Administration information

Name	Form number	PDF file name
<i>IBM DB2 Universal Database Administration Guide: Planning</i>	SC09-4822	db2d1x80
<i>IBM DB2 Universal Database Administration Guide: Implementation</i>	SC09-4820	db2d2x80
<i>IBM DB2 Universal Database Administration Guide: Performance</i>	SC09-4821	db2d3x80
<i>IBM DB2 Universal Database Administrative API Reference</i>	SC09-4824	db2b0x80

Table 19. Administration information (continued)

Name	Form number	PDF file name
<i>IBM DB2 Universal Database Data Movement Utilities Guide and Reference</i>	SC09-4830	db2dmx80
<i>IBM DB2 Universal Database Data Recovery and High Availability Guide and Reference</i>	SC09-4831	db2hax80
<i>IBM DB2 Universal Database Data Warehouse Center Administration Guide</i>	SC27-1123	db2ddx80
<i>IBM DB2 Universal Database Federated Systems Guide</i>	GC27-1224	db2fpx80
<i>IBM DB2 Universal Database Guide to GUI Tools for Administration and Development</i>	SC09-4851	db2atx80
<i>IBM DB2 Universal Database Replication Guide and Reference</i>	SC27-1121	db2e0x80
<i>IBM DB2 Installing and Administering a Satellite Environment</i>	GC09-4823	db2dsx80
<i>IBM DB2 Universal Database SQL Reference, Volume 1</i>	SC09-4844	db2s1x80
<i>IBM DB2 Universal Database SQL Reference, Volume 2</i>	SC09-4845	db2s2x80
<i>IBM DB2 Universal Database System Monitor Guide and Reference</i>	SC09-4847	db2f0x80

Application development information

The information in this category is of special interest to application developers or programmers working with DB2. You will find information about supported languages and compilers, as well as the documentation required to access DB2 using the various supported programming interfaces, such as embedded SQL, ODBC, JDBC, SQLj, and CLI. If you view this information online in HTML you can also access a set of DB2 sample programs in HTML.

The installation directory for this category is doc/htmlcd/%L/ad.

Table 20. Application development information

Name	Form number	PDF file name
<i>IBM DB2 Universal Database Application Development Guide: Building and Running Applications</i>	SC09-4825	db2axx80
<i>IBM DB2 Universal Database Application Development Guide: Programming Client Applications</i>	SC09-4826	db2a1x80
<i>IBM DB2 Universal Database Application Development Guide: Programming Server Applications</i>	SC09-4827	db2a2x80
<i>IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 1</i>	SC09-4849	db2l1x80
<i>IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 2</i>	SC09-4850	db2l2x80
<i>IBM DB2 Universal Database Data Warehouse Center Application Integration Guide</i>	SC27-1124	db2adx80
<i>IBM DB2 XML Extender Administration and Programming</i>	SC27-1234	db2sxx80

Business intelligence information

The information in this category describes how to use components that enhance the data warehousing and analytical capabilities of DB2 Universal Database.

The installation directory for this category is doc/htmlcd/%L/wareh.

Table 21. Business intelligence information

Name	Form number	PDF file name
<i>IBM DB2 Warehouse Manager Information Catalog Center Administration Guide</i>	SC27-1125	db2dix80
<i>IBM DB2 Warehouse Manager Installation Guide</i>	GC27-1122	db2idx80

DB2 Connect information

The information in this category describes how to access host or iSeries data using DB2 Connect Enterprise Edition or DB2 Connect Personal Edition.

The installation directory for this category is doc/htmlcd/%L/conn.

Table 22. DB2 Connect information

Name	Form number	PDF file name
<i>APPC, CPI-C, and SNA Sense Codes</i>	No form number	db2apx80
<i>IBM Connectivity Supplement</i>	No form number	db2h1x80
<i>IBM DB2 Connect Quick Beginnings for DB2 Connect Enterprise Edition</i>	GC09-4833	db2c6x80
<i>IBM DB2 Connect Quick Beginnings for DB2 Connect Personal Edition</i>	GC09-4834	db2c1x80
<i>IBM DB2 Connect User's Guide</i>	SC09-4835	db2c0x80

Getting started information

The information in this category is useful when you are installing and configuring servers, clients, and other DB2 products.

The installation directory for this category is doc/htmlcd/%L/start.

Table 23. Getting started information

Name	Form number	PDF file name
<i>IBM DB2 Universal Database Quick Beginnings for DB2 Clients</i>	GC09-4832	db2itx80
<i>IBM DB2 Universal Database Quick Beginnings for DB2 Servers</i>	GC09-4836	db2isx80
<i>IBM DB2 Universal Database Quick Beginnings for DB2 Personal Edition</i>	GC09-4838	db2i1x80
<i>IBM DB2 Universal Database Installation and Configuration Supplement</i>	GC09-4837	db2iyx80
<i>IBM DB2 Universal Database Quick Beginnings for DB2 Data Links Manager</i>	GC09-4829	db2z6x80

Tutorial information

Tutorial information introduces DB2 features and teaches how to perform various tasks.

The installation directory for this category is `doc/htmlcd/%L/tutr`.

Table 24. Tutorial information

Name	Form number	PDF file name
<i>Business Intelligence Tutorial: Introduction to the Data Warehouse</i>	No form number	db2tux80
<i>Business Intelligence Tutorial: Extended Lessons in Data Warehousing</i>	No form number	db2tax80
<i>Development Center Tutorial for Video Online using Microsoft Visual Basic</i>	No form number	db2tdx80
<i>Information Catalog Center Tutorial</i>	No form number	db2aix80
<i>Video Central for e-business Tutorial</i>	No form number	db2twx80
<i>Visual Explain Tutorial</i>	No form number	db2tvx80

Optional component information

The information in this category describes how to work with optional DB2 components.

The installation directory for this category is `doc/htmlcd/%L/opt`.

Table 25. Optional component information

Name	Form number	PDF file name
<i>IBM DB2 Life Sciences Data Connect Planning, Installation, and Configuration Guide</i>	GC27-1235	db2lsx80
<i>IBM DB2 Spatial Extender User's Guide and Reference</i>	SC27-1226	db2sbx80
<i>IBM DB2 Universal Database Data Links Manager Administration Guide and Reference</i>	SC27-1221	db2z0x80

Table 25. Optional component information (continued)

Name	Form number	PDF file name
IBM DB2 Universal Database Net Search Extender Administration and Programming Guide	SH12-6740	N/A
Note: HTML for this document is not installed from the HTML documentation CD.		

Release notes

The release notes provide additional information specific to your product's release and FixPak level. They also provides summaries of the documentation updates incorporated in each release and FixPak.

Table 26. Release notes

Name	Form number	PDF file name	HTML directory
DB2 Release Notes	See note.	See note.	doc/prodcd/%L/db2ir where %L is the language identifier.
DB2 Connect Release Notes	See note.	See note.	doc/prodcd/%L/db2cr where %L is the language identifier.
DB2 Installation Notes	Available on product CD-ROM only.	Available on product CD-ROM only.	

Note: The HTML version of the release notes is available from the Information Center and on the product CD-ROMs. To view the ASCII file:

- On UNIX-based platforms, see the Release.Notes file. This file is located in the DB2DIR/Readme/%L directory, where %L represents the locale name and DB2DIR represents:
 - /usr/opt/db2_08_01 on AIX
 - /opt/IBM/db2/V8.1 on all other UNIX operating systems
- On other platforms, see the RELEASE.TXT file. This file is located in the directory where the product is installed.

Related tasks:

- “Printing DB2 books from PDF files” on page 393

- “Ordering printed DB2 books” on page 394
- “Accessing online help” on page 394
- “Finding product information by accessing the DB2 Information Center from the administration tools” on page 398
- “Viewing technical documentation online directly from the DB2 HTML Documentation CD” on page 399

Printing DB2 books from PDF files

You can print DB2 books from the PDF files on the *DB2 PDF Documentation* CD. Using Adobe Acrobat Reader, you can print either the entire book or a specific range of pages.

Prerequisites:

Ensure that you have Adobe Acrobat Reader. It is available from the Adobe Web site at www.adobe.com

Procedure:

To print a DB2 book from a PDF file:

1. Insert the *DB2 PDF Documentation* CD. On UNIX operating systems, mount the DB2 PDF Documentation CD. Refer to your *Quick Beginnings* book for details on how to mount a CD on UNIX operating systems.
2. Start Adobe Acrobat Reader.
3. Open the PDF file from one of the following locations:
 - On Windows operating systems:
x:\doc\language directory, where *x* represents the CD-ROM drive letter and *language* represents the two-character territory code that represents your language (for example, EN for English).
 - On UNIX operating systems:
/cdrom/doc/%L directory on the CD-ROM, where */cdrom* represents the mount point of the CD-ROM and *%L* represents the name of the desired locale.

Related tasks:

- “Ordering printed DB2 books” on page 394
- “Finding product information by accessing the DB2 Information Center from the administration tools” on page 398
- “Viewing technical documentation online directly from the DB2 HTML Documentation CD” on page 399

Related reference:

- “Overview of DB2 Universal Database technical information” on page 385

Ordering printed DB2 books

Procedure:

To order printed books:

- Contact your IBM authorized dealer or marketing representative. To find a local IBM representative, check the IBM Worldwide Directory of Contacts at www.ibm.com/shop/planetwide
- Phone 1-800-879-2755 in the United States or 1-800-IBM-4YOU in Canada.
- Visit the IBM Publications Center at www.ibm.com/shop/publications/order

Related tasks:

- “Printing DB2 books from PDF files” on page 393
- “Finding topics by accessing the DB2 Information Center from a browser” on page 396
- “Viewing technical documentation online directly from the DB2 HTML Documentation CD” on page 399

Related reference:

- “Overview of DB2 Universal Database technical information” on page 385

Accessing online help

The online help that comes with all DB2 components is available in three types:

- Window and notebook help
- Command line help
- SQL statement help

Window and notebook help explain the tasks that you can perform in a window or notebook and describe the controls. This help has two types:

- Help accessible from the **Help** button
- Infopops

The **Help** button gives you access to overview and prerequisite information. The infopops describe the controls in the window or notebook. Window and notebook help are available from DB2 centers and components that have user interfaces.

Command line help includes Command help and Message help. Command help explains the syntax of commands in the command line processor. Message help describes the cause of an error message and describes any action you should take in response to the error.

SQL statement help includes SQL help and SQLSTATE help. DB2 returns an SQLSTATE value for conditions that could be the result of an SQL statement. SQLSTATE help explains the syntax of SQL statements (SQL states and class codes).

Note: SQL help is not available for UNIX operating systems.

Procedure:

To access online help:

- For window and notebook help, click **Help** or click that control, then click **F1**. If the **Automatically display infopops** check box on the **General** page of the **Tool Settings** notebook is selected, you can also see the infopop for a particular control by holding the mouse cursor over the control.
- For command line help, open the command line processor and enter:

- For Command help:

? command

where *command* represents a keyword or the entire command.

For example, *? catalog* displays help for all the CATALOG commands, while *? catalog database* displays help for the CATALOG DATABASE command.

- For Message help:

? XXXnnnnn

where *XXXnnnnn* represents a valid message identifier.

For example, *? SQL30081* displays help about the SQL30081 message.

- For SQL statement help, open the command line processor and enter:

- For SQL help:

? sqlstate or *? class code*

where *sqlstate* represents a valid five-digit SQL state and *class code* represents the first two digits of the SQL state.

For example, *? 08003* displays help for the 08003 SQL state, while *? 08* displays help for the 08 class code.

- For SQLSTATE help:

`help statement`

where *statement* represents an SQL statement.

For example, `help SELECT` displays help about the `SELECT` statement.

Related tasks:

- “Finding topics by accessing the DB2 Information Center from a browser” on page 396
- “Viewing technical documentation online directly from the DB2 HTML Documentation CD” on page 399

Finding topics by accessing the DB2 Information Center from a browser

The DB2 Information Center accessed from a browser enables you to access the information you need to take full advantage of DB2 Universal Database and DB2 Connect. The DB2 Information Center also documents major DB2 features and components including replication, data warehousing, metadata, Life Sciences Data Connect, and DB2 extenders.

The DB2 Information Center accessed from a browser is composed of the following major elements:

Navigation tree

The navigation tree is located in the left frame of the browser window. The tree expands and collapses to show and hide topics, the glossary, and the master index in the DB2 Information Center.

Navigation toolbar

The navigation toolbar is located in the top right frame of the browser window. The navigation toolbar contains buttons that enable you to search the DB2 Information Center, hide the navigation tree, and find the currently displayed topic in the navigation tree.

Content frame

The content frame is located in the bottom right frame of the browser window. The content frame displays topics from the DB2 Information Center when you click on a link in the navigation tree, click on a search result, or follow a link from another topic or from the master index.

Prerequisites:

To access the DB2 Information Center from a browser, you must use one of the following browsers:

- Microsoft Explorer, version 5 or later
- Netscape Navigator, version 6.1 or later

Restrictions:

The DB2 Information Center contains only those sets of topics that you chose to install from the *DB2 HTML Documentation CD*. If your Web browser returns a File not found error when you try to follow a link to a topic, you must install one or more additional sets of topics *DB2 HTML Documentation CD*.

Procedure:

To find a topic by searching with keywords:

1. In the navigation toolbar, click **Search**.
2. In the top text entry field of the Search window, enter two or more terms related to your area of interest and click **Search**. A list of topics ranked by accuracy displays in the **Results** field.

Entering more terms increases the precision of your query while reducing the number of topics returned from your query.

3. In the **Results** field, click the title of the topic you want to read. The topic displays in the content frame.

To find a topic in the navigation tree:

1. In the navigation tree, click the book icon of the category of topics related to your area of interest. A list of subcategories displays underneath the icon.
2. Continue to click the book icons until you find the category containing the topics in which you are interested. Categories that link to topics display the category title as an underscored link when you move the cursor over the category title. The navigation tree identifies topics with a page icon.
3. Click the topic link. The topic displays in the content frame.

To find a topic or term in the master index:

1. In the navigation tree, click the "Index" category. The category expands to display a list of links arranged in alphabetical order in the navigation tree.
2. In the navigation tree, click the link corresponding to the first character of the term relating to the topic in which you are interested. A list of terms with that initial character displays in the content frame. Terms that have multiple index entries are identified by a book icon.
3. Click the book icon corresponding to the term in which you are interested. A list of subterms and topics displays below the term you clicked. Topics are identified by page icons with an underscored title.
4. Click on the title of the topic that meets your needs. The topic displays in the content frame.

Related concepts:

- “Accessibility” on page 405
- “DB2 Information Center for topics” on page 407

Related tasks:

- “Finding product information by accessing the DB2 Information Center from the administration tools” on page 398
- “Updating the HTML documentation installed on your machine” on page 400
- “Troubleshooting DB2 documentation search with Netscape 4.x” on page 402
- “Searching the DB2 documentation” on page 403

Related reference:

- “Overview of DB2 Universal Database technical information” on page 385

Finding product information by accessing the DB2 Information Center from the administration tools

The DB2 Information Center provides quick access to DB2 product information and is available on all operating systems for which the DB2 administration tools are available.

The DB2 Information Center accessed from the tools provides six types of information.

Tasks Key tasks you can perform using DB2.

Concepts

Key concepts for DB2.

Reference

DB2 reference information, such as keywords, commands, and APIs.

Troubleshooting

Error messages and information to help you with common DB2 problems.

Samples

Links to HTML listings of the sample programs provided with DB2.

Tutorials

Instructional aid designed to help you learn a DB2 feature.

Prerequisites:

Some links in the DB2 Information Center point to Web sites on the Internet. To display the content for these links, you will first have to connect to the Internet.

Procedure:

To find product information by accessing the DB2 Information Center from the tools:

1. Start the DB2 Information Center in one of the following ways:
 - From the graphical administration tools, click on the **Information Center** icon in the toolbar. You can also select it from the **Help** menu.
 - At the command line, enter **db2ic**.
2. Click the tab of the information type related to the information you are attempting to find.
3. Navigate through the tree and click on the topic in which you are interested. The Information Center will then launch a Web browser to display the information.
4. To find information without browsing the lists, click the **Search** icon to the right of the list.

Once the Information Center has launched a browser to display the information, you can perform a full-text search by clicking the **Search** icon in the navigation toolbar.

Related concepts:

- “Accessibility” on page 405
- “DB2 Information Center for topics” on page 407

Related tasks:

- “Finding topics by accessing the DB2 Information Center from a browser” on page 396
- “Searching the DB2 documentation” on page 403

Viewing technical documentation online directly from the DB2 HTML Documentation CD

All of the HTML topics that you can install from the *DB2 HTML Documentation CD* can also be read directly from the CD. Therefore, you can view the documentation without having to install it.

Restrictions:

Because the following items are installed from the DB2 product CD and not the *DB2 HTML Documentation CD*, you must install the DB2 product to view these items:

- Tools help
- DB2 Quick Tour
- Release notes

Procedure:

1. Insert the *DB2 HTML Documentation CD*. On UNIX operating systems, mount the *DB2 HTML Documentation CD*. Refer to your *Quick Beginnings* book for details on how to mount a CD on UNIX operating systems.
2. Start your HTML browser and open the appropriate file:

- For Windows operating systems:

```
e:\Program Files\sql11ib\doc\htmlcd\%L\index.htm
```

where *e* represents the CD-ROM drive, and %L is the locale of the documentation that you wish to use, for example, **en_US** for English.

- For UNIX operating systems:

```
/cdrom/Program Files/sql11ib/doc/htmlcd/%L/index.htm
```

where */cdrom/* represents where the CD is mounted, and %L is the locale of the documentation that you wish to use, for example, **en_US** for English.

Related tasks:

- “Finding topics by accessing the DB2 Information Center from a browser” on page 396
- “Copying files from the DB2 HTML Documentation CD to a Web Server” on page 402

Related reference:

- “Overview of DB2 Universal Database technical information” on page 385

Updating the HTML documentation installed on your machine

It is now possible to update the HTML installed from the *DB2 HTML Documentation CD* when updates are made available from IBM. This can be done in one of two ways:

- Using the Information Center (if you have the DB2 administration GUI tools installed).
- By downloading and applying a DB2 HTML documentation FixPak .

Note: This will NOT update the DB2 code; it will only update the HTML documentation installed from the *DB2 HTML Documentation CD*.

Procedure:

To use the Information Center to update your local documentation:

1. Start the DB2 Information Center in one of the following ways:
 - From the graphical administration tools, click on the **Information Center** icon in the toolbar. You can also select it from the **Help** menu.
 - At the command line, enter **db2ic**.
2. Ensure your machine has access to the external Internet; the updater will download the latest documentation FixPak from the IBM server if required.
3. Select **Information Center** —> **Update Local Documentation** from the menu to start the update.
4. Supply your proxy information (if required) to connect to the external Internet.

The Information Center will download and apply the latest documentation FixPak, if one is available.

To manually download and apply the documentation FixPak :

1. Ensure your machine is connected to the Internet.
2. Open the DB2 support page in your Web browser at:
www.ibm.com/software/data/db2/udb/winos2unix/support
3. Follow the link for version 8 and look for the "Documentation FixPaks" link.
4. Determine if the version of your local documentation is out of date by comparing the documentation FixPak level to the documentation level you have installed. This current documentation on your machine is at the following level: **DB2 v8.1 GA**.
5. If there is a more recent version of the documentation available then download the FixPak applicable to your operating system. There is one FixPak for all Windows platforms, and one FixPak for all UNIX platforms.
6. Apply the FixPak:
 - For Windows operating systems: The documentation FixPak is a self extracting zip file. Place the downloaded documentation FixPak in an empty directory, and run it. It will create a **setup** command which you can run to install the documentation FixPak.
 - For UNIX operating systems: The documentation FixPak is a compressed tar.Z file. Uncompress and untar the file. It will create a directory named `delta_install` with a script called **installdocfix**. Run this script to install the documentation FixPak.

Related tasks:

- “Copying files from the DB2 HTML Documentation CD to a Web Server” on page 402

Related reference:

- “Overview of DB2 Universal Database technical information” on page 385

Copying files from the DB2 HTML Documentation CD to a Web Server

The entire DB2 information library is delivered to you on the *DB2 HTML Documentation CD*, so you can install the library on a Web server for easier access. Simply copy to your Web server the documentation for the languages that you want.

Procedure:

To copy files from the *DB2 HTML Documentation CD* to a Web server, use the appropriate path:

- For Windows operating systems:

```
E:\Program Files\sqllib\doc\htmlcd\%L\*.*
```

where *E* represents the CD-ROM drive and *%L* represents the language identifier.

- For UNIX operating systems:

```
/cdrom:Program Files/sqllib/doc/htmlcd/%L/*.*
```

where *cdrom* represents the CD-ROM drive and *%L* represents the language identifier.

Related tasks:

- “Searching the DB2 documentation” on page 403

Related reference:

- “Supported DB2 interface languages, locales, and code pages” in the *Quick Beginnings for DB2 Servers*
- “Overview of DB2 Universal Database technical information” on page 385

Troubleshooting DB2 documentation search with Netscape 4.x

Most search problems are related to the Java support provided by web browsers. This task describes possible workarounds.

Procedure:

A common problem with Netscape 4.x involves a missing or misplaced security class. Try the following workaround, especially if you see the following line in the browser Java console:

```
Cannot find class java/security/InvalidParameterException
```

- On Windows operating systems:

From the *DB2 HTML Documentation CD*, copy the supplied `x:Program Files\sql1lib\doc\htmlcd\locale\InvalidParameterException.class` file to the `java\classes\java\security\` directory relative to your Netscape browser installation, where *x* represents the CD-ROM drive letter and *locale* represents the name of the desired locale.

Note: You may have to create the `java\security\` subdirectory structure.

- On UNIX operating systems:

From the *DB2 HTML Documentation CD*, copy the supplied `/cdrom/Program Files/sql1lib/doc/htmlcd/locale/InvalidParameterException.class` file to the `java/classes/java/security/` directory relative to your Netscape browser installation, where *cdrom* represents the mount point of the CD-ROM and *locale* represents the name of the desired locale.

Note: You may have to create the `java/security/` subdirectory structure.

If your Netscape browser still fails to display the search input window, try the following:

- Stop all instances of Netscape browsers to ensure that there is no Netscape code running on the machine. Then open a new instance of the Netscape browser and try to start the search again.
- Purge the browser's cache.
- Try a different version of Netscape, or a different browser.

Related tasks:

- "Searching the DB2 documentation" on page 403

Searching the DB2 documentation

To search DB2's documentation, you need Netscape 6.1 or higher, or Microsoft's Internet Explorer 5 or higher. Ensure that your browser's Java support is enabled.

A pop-up search window opens when you click the search icon in the navigation toolbar of the Information Center accessed from a browser. If you are using the search for the first time it may take a minute or so to load into the search window.

Restrictions:

The following restrictions apply when you use the documentation search:

- Boolean searches are not supported. The boolean search qualifiers *and* and *or* will be ignored in a search. For example, the following searches would produce the same results:
 - servlets *and* beans
 - servlets *or* beans
- Wildcard searches are not supported. A search on *java** will only look for the literal string *java** and would not, for example, find *javadoc*.

In general, you will get better search results if you search for phrases instead of single words.

Procedure:

To search the DB2 documentation:

1. In the navigation toolbar, click **Search**.
2. In the top text entry field of the Search window, enter two or more terms related to your area of interest and click **Search**. A list of topics ranked by accuracy displays in the **Results** field.
Entering more terms increases the precision of your query while reducing the number of topics returned from your query.
3. In the **Results** field, click the title of the topic you want to read. The topic displays in the content frame.

Note: When you perform a search, the first result is automatically loaded into your browser frame. To view the contents of other search results, click on the result in results lists.

Related tasks:

- “Troubleshooting DB2 documentation search with Netscape 4.x” on page 402

Online DB2 troubleshooting information

With the release of DB2[®] UDB Version 8, there will no longer be a *Troubleshooting Guide*. The troubleshooting information once contained in this guide has been integrated into the DB2 publications. By doing this, we are able to deliver the most up-to-date information possible. To find information on the troubleshooting utilities and functions of DB2, access the DB2 Information Center from any of the tools.

Refer to the DB2 Online Support site if you are experiencing problems and want help finding possible causes and solutions. The support site contains a

large, constantly updated database of DB2 publications, TechNotes, APAR (product problem) records, FixPaks, and other resources. You can use the support site to search through this knowledge base and find possible solutions to your problems.

Access the Online Support site at www.ibm.com/software/data/db2/udb/winos2unix/support, or by clicking the **Online Support** button in the DB2 Information Center. Frequently changing information, such as the listing of internal DB2 error codes, is now also available from this site.

Related concepts:

- “DB2 Information Center for topics” on page 407

Related tasks:

- “Finding product information by accessing the DB2 Information Center from the administration tools” on page 398

Accessibility

Accessibility features help users with physical disabilities, such as restricted mobility or limited vision, to use software products successfully. These are the major accessibility features in DB2[®] Universal Database Version 8:

- DB2 allows you to operate all features using the keyboard instead of the mouse. See “Keyboard Input and Navigation”.
- DB2 enables you customize the size and color of your fonts. See “Accessible Display” on page 406.
- DB2 allows you to receive either visual or audio alert cues. See “Alternative Alert Cues” on page 406.
- DB2 supports accessibility applications that use the Java[™] Accessibility API. See “Compatibility with Assistive Technologies” on page 406.
- DB2 comes with documentation that is provided in an accessible format. See “Accessible Documentation” on page 406.

Keyboard Input and Navigation

Keyboard Input

You can operate the DB2 Tools using only the keyboard. You can use keys or key combinations to perform most operations that can also be done using a mouse.

Keyboard Focus

In UNIX-based systems, the position of the keyboard focus is highlighted, indicating which area of the window is active and where your keystrokes will have an effect.

Accessible Display

The DB2 Tools have features that enhance the user interface and improve accessibility for users with low vision. These accessibility enhancements include support for customizable font properties.

Font Settings

The DB2 Tools allow you to select the color, size, and font for the text in menus and dialog windows, using the Tools Settings notebook.

Non-dependence on Color

You do not need to distinguish between colors in order to use any of the functions in this product.

Alternative Alert Cues

You can specify whether you want to receive alerts through audio or visual cues, using the Tools Settings notebook.

Compatibility with Assistive Technologies

The DB2 Tools interface supports the Java Accessibility API enabling use by screen readers and other assistive technologies used by people with disabilities.

Accessible Documentation

Documentation for the DB2 family of products is available in HTML format. This allows you to view documentation according to the display preferences set in your browser. It also allows you to use screen readers and other assistive technologies.

DB2 tutorials

The DB2® tutorials help you learn about various aspects of DB2 Universal Database. The tutorials provide lessons with step-by-step instructions in the areas of developing applications, tuning SQL query performance, working with data warehouses, managing metadata, and developing Web services using DB2.

Before you begin:

Before you can access these tutorials using the links below, you must install the tutorials from the *DB2 HTML Documentation* CD-ROM.

If you do not want to install the tutorials, you can view the HTML versions of the tutorials directly from the *DB2 HTML Documentation CD*. PDF versions of these tutorials are also available on the *DB2 PDF Documentation CD*.

Some tutorial lessons use sample data or code. See each individual tutorial for a description of any prerequisites for its specific tasks.

DB2 Universal Database tutorials:

If you installed the tutorials from the *DB2 HTML Documentation CD-ROM*, you can click on a tutorial title in the following list to view that tutorial.

Business Intelligence Tutorial: Introduction to the Data Warehouse Center
Perform introductory data warehousing tasks using the Data Warehouse Center.

Business Intelligence Tutorial: Extended Lessons in Data Warehousing
Perform advanced data warehousing tasks using the Data Warehouse Center.

Development Center Tutorial for Video Online using Microsoft® Visual Basic
Build various components of an application using the Development Center Add-in for Microsoft Visual Basic.

Information Catalog Center Tutorial
Create and manage an information catalog to locate and use metadata using the Information Catalog Center.

Video Central for e-business Tutorial
Develop and deploy an advanced DB2 Web Services application using WebSphere® products.

Visual Explain Tutorial
Analyze, optimize, and tune SQL statements for better performance using Visual Explain.

DB2 Information Center for topics

The DB2® Information Center gives you access to all of the information you need to take full advantage of DB2 Universal Database™ and DB2 Connect™ in your business. The DB2 Information Center also documents major DB2 features and components including replication, data warehousing, the Information Catalog Center, Life Sciences Data Connect, and DB2 extenders.

The DB2 Information Center accessed from a browser has the following features:

Regularly updated documentation

Keep your topics up-to-date by downloading updated HTML.

Search

Search all of the topics installed on your workstation by clicking **Search** in the navigation toolbar.

Integrated navigation tree

Locate any topic in the DB2 library from a single navigation tree. The navigation tree is organized by information type as follows:

- Tasks provide step-by-step instructions on how to complete a goal.
- Concepts provide an overview of a subject.
- Reference topics provide detailed information about a subject, including statement and command syntax, message help, requirements.

Master index

Access the information in topics and tools help from one master index. The index is organized in alphabetical order by index term.

Master glossary

The master glossary defines terms used in the DB2 Information Center. The glossary is organized in alphabetical order by glossary term.

Related tasks:

- “Finding topics by accessing the DB2 Information Center from a browser” on page 396
- “Finding product information by accessing the DB2 Information Center from the administration tools” on page 398
- “Updating the HTML documentation installed on your machine” on page 400

Appendix K. Notices

IBM may not offer the products, services, or features discussed in this document in all 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.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country/region or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country/region 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.

Licenses of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information that has been exchanged, should contact:

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems, and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information may contain 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 may contain sample application programs, in source language, which illustrate 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.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (*your company name*) (*year*). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. *_enter the year or years_*. All rights reserved.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both, and have been used in at least one of the documents in the DB2 UDB documentation library.

ACF/VTAM	LAN Distance
AISPO	MVS
AIX	MVS/ESA
AIXwindows	MVS/XA
AnyNet	Net.Data
APPN	NetView
AS/400	OS/390
BookManager	OS/400
C Set++	PowerPC
C/370	pSeries
CICS	QBIC
Database 2	QMF
DataHub	RACF
DataJoiner	RISC System/6000
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/400
DB2 Extenders	SQL/DS
DB2 OLAP Server	System/370
DB2 Universal Database	System/390
Distributed Relational Database Architecture	SystemView
DRDA	Tivoli
eServer	VisualAge
Extended Services	VM/ESA
FFST	VSE/ESA
First Failure Support Technology	VTAM
IBM	WebExplorer
IMS	WebSphere
IMS/ESA	WIN-OS/2
iSeries	z/OS
	zSeries

The following terms are trademarks or registered trademarks of other companies and have been used in at least one of the documents in the DB2 UDB documentation library:

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

Intel and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

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

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

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

Index

A

- accessibility
 - features 405
- AIX
 - creating a DB2 home file system for a partitioned database system 76
 - creating required users for a partitioned DB2 server installation 78
 - installation requirements for DB2 servers 55
 - installing a partitioned DB2 server 67
 - installing DB2 servers 54
 - mounting the DB2 CD-ROM 58, 80
 - updating environment settings 73
 - verifying that NFS is running 75
- application development
 - binding utilities 280
 - cataloging satellite control database 279
 - DB2 Synchronizer
 - application 290
 - db2GetSyncSession API 283
 - db2QuerySatelliteProgress API 287
 - db2SetSyncSession API 282
 - db2SyncSatellite API 287
 - db2SyncSatelliteStop API 288
 - db2SyncSatelliteTest API 284
 - overview 278
 - synchronization overview 285
 - application programming interfaces
 - db2GetSyncSession 283
 - db2QuerySatelliteProgress 287
 - db2SetSyncSession 282
 - db2SyncSatellite 287
 - db2SyncSatelliteStop 288
 - db2SyncSatelliteTest 284
 - synchronization overview 285
 - application versions
 - changing batch step of group batches 260
 - creating 196
 - creating for test of group batches 257
 - application versions (*continued*)
 - creating level 0 for test of group batches 258
 - creating test level from
 - production level 215
 - db2GetSyncSession API 283
 - db2SetSyncSession API 282
 - editing level 0 for test of group batches 259
 - example test-production cycle 204
 - level 0 in test state fully modifiable 217
 - levels 206
 - life cycle 201
 - locking of batch steps 217
 - maximum length 282
 - not defined in satellite control database 334
 - not set at satellite 335
 - obsolete level 208
 - obsoleting production level 216
 - overview xx
 - parameterizing script 193
 - production level 208
 - promoting test level to production level 214
 - purpose 196
 - recorded for satellite on first synchronization 263
 - relationship between batch state and batch step 217
 - response file consideration 313
 - retrieving from satellite 196
 - satellite synchronization
 - component must be installed 249
 - setting up satellite for synchronization test 249
 - states 213
 - states of level 208
 - test level 208
 - testing group batches with model office 273
 - update batch in test level 213
 - use during synchronization 196
 - applications
 - applications (*continued*)
 - associated with application version 196
 - binding and running
 - synchronizing application 289
 - installing new version
 - creating new application version for group 322
 - creating system for test deployment 323
 - deploying new application to production satellites 325
 - determining application version of satellite 325
 - overview 320
 - setting new application version on satellite 321
 - synchronizing
 - overview 277
 - assigning fix batch to satellite 340
 - authentication credentials
 - changing password on satellite control server 225
 - changing password on target server 226
 - creating 222, 255
 - creating satadmin.aut file on satellite 223
 - description 188
 - finding for execution target 226
 - finding for group 225
 - interrupted synchronization session can result in
 - authentication error 226
 - not available on satellite 331
 - overview xx
 - password encrypted 221
 - purpose 221
 - recreating satadmin.aut file on satellite 346
 - setting for synchronization using response file 313
 - setting up for synchronization test 242
 - storage on satellite 223
 - stored at satellite control server 222
 - updating satadmin.aut file on satellite 346

availability of satellite control server 296

B

batch steps

- authentication credential component 188
- authentication of script 224
- cataloging execution target 229
- changing 260
- components 188
- error return codes 345
- execution target component 188
- locking of 217
- overview xx
- parameterizing script 193
- purpose 188
- relationship with batch state 217
- script component 188
- setting execution point to next batch step 182
- success code set component 188
- test
 - execution by test satellites 200
 - promoting to production 201
- unlocked for test-level update batch 213

batches

- assigning fix batch to satellite 340
- batch step components 188
- cataloging execution target 229
- creating fix batch using model office 275
- debugging fix batch 342
- execution logs on satellite 346
- fix 219
- modes 194
- overview xx
- purpose 187
- relationship between state and batch steps 217
- setting execution starting point to next batch step 182
- states of application-version level 208

bind files

- granting privileges on 247

binding utilities 280

building applications 278

C

cataloging

- instances and databases on production satellites 237
- instances and databases on test satellites 236
- instances and databases, requirement 230
- local instance on model office 234
- model office on Control Center 231
- remote instances and databases on model office 232
- remote instances and databases on model office using client profile 233
- satellite control server and satellite control database on Control Center 231

- cleanup batches 197
- client profile
 - test satellite, setting up 237
 - using to perform cataloging on model office 233
- command line processor (CLP)
 - verifying the installation 26, 61, 108

commands

- cpyssetup.bat 310
- db2cfexp 235
- db2cfimp 233
- db2rbind 175
- db2rspgn 237
- set db2instance 9

completing a mass deployment 320

completing setup of model office 235

Configuration Assistant

- creating client profile 233

contextual parameters 193

Control Center

- backing up directories 295
- cataloging instances and databases
 - overview 230
 - procedure 231
- cataloging model office 231

instance 229

- satellite control server and satellite control database must be cataloged on 229

copying DB2 installation files to media for mass installation 314

copying response file and client profile files to media for mass installation 314

creating

- DB2CTLSV instance 7
- SATCTLDB database 8

creating test level from production 215

customizing response file response file 313

D

database definition xiii

database partition servers

- enabling communications 89
- installing on Windows 45

database recovery

- managing backup images 306
- model office 298
- model office during development phase 299
- production satellites 306
- test environment 298
- test satellite 298
- test satellites 298
- types of 298

databases

- backing up before DB2 migration 147, 160
- creating using response file 313
- migrating 151, 153, 162, 167
- SATCTLDB
 - general administration tables 357

DB2

installing

- applying the latest FixPak 25, 48, 61, 92, 108
- verifying the installation 26, 61, 108

migrating

- backing up databases 147, 160
- changing the diagnostic error level 150
- overview 153, 167
- recommendations 146, 158
- restrictions 144, 156
- space considerations 148, 161
- system configuration settings 149, 162
- taking servers offline 152, 164

- DB2 (*continued*)
 - partitioned installation
 - updating AIX environment settings 73
 - DB2 Administration Server (DAS)
 - migrating 167
 - DB2 CD-ROM
 - copying to your computer 80
 - mounting
 - on AIX 58, 80
 - DB2 control server
 - response file keywords
 - for Windows 122
 - DB2 documentation 27, 50, 109
 - installing on UNIX 62, 94
 - installing on Windows 27, 50, 109
 - DB2 documentation search
 - using Netscape 4.x 402
 - DB2 Enterprise Server Edition
 - installing
 - Windows 37
 - DB2 ESE/WSE
 - disk requirements
 - on UNIX 57
 - DB2 home file system
 - partitioned database system, AIX 76
 - DB2 Information Center 407
 - DB2 install file
 - importing into SMS 136
 - DB2 installation package
 - distributing
 - across the network 138
 - DB2 objects
 - naming rules 375
 - DB2 Personal Edition
 - disk requirements
 - Windows 102
 - installing
 - user accounts 104
 - Windows 98, 100
 - installing on Windows 97
 - memory requirements
 - Windows 102
 - migrating
 - Windows 169
 - migrating databases
 - Windows 173
 - preparing to migrate
 - Windows 170
 - setting up user accounts 104
 - DB2 processes
 - killing
 - during a response file installation 126
 - during an interactive installation 125
 - DB2 products
 - installing
 - using SMS 135
 - DB2 servers
 - installation requirements
 - AIX 55
 - for partitions on AIX 69
 - Windows 17
 - installing
 - Windows 12, 14
 - installing on AIX 54
 - installing on UNIX 52
 - installing on Windows 23
 - memory requirements
 - UNIX 71
 - Windows 19
 - partitioned
 - installing on Windows 33
 - installing on AIX 67
 - installing on UNIX 64
 - installing on Windows 29, 30
 - preparing the Windows environment 37
 - taking offline 152, 164
 - DB2 Setup wizard
 - installing database partition server
 - UNIX 81
 - installing database partition servers 45
 - installing DB2 servers
 - UNIX 58
 - starting on Windows 105
 - DB2 Synchronizer 290
 - DB2 trace facility (db2trc)
 - running on satellite 344
 - DB2 tutorials 406
 - DB2 Universal Database
 - response file
 - keywords 121
 - db2cli.ini file
 - configuring for a response file installation 142
 - DB2CTLSV instance
 - creating 7
 - db2GetSyncSession API 283
 - db2QuerySatelliteProgress API 287
 - db2rbind command 175
 - db2rspgn response file
 - generator 124
 - db2satcs stored procedure
 - rebinding to SATCTLDB database 336
 - DB2SATELLITEID registry
 - variable 250
 - setting up satellite for
 - synchronization test 250
 - db2SetSyncSession API 282
 - db2SyncSatellite API 287
 - db2SyncSatelliteTest API 284
 - db2trc (DB2 trace facility)
 - running on satellite 344
 - debugging
 - fix batch 342
 - debugging test batches 265
 - deploying a new group
 - changing batch steps 260
 - setting execution starting point for a satellite 267
 - development and acceptance-testing
 - phase, model office consideration 269
 - diagnostic error level
 - changing before DB2 migration 150
 - directories
 - setting up shared access 131
 - directory schema
 - extending
 - on Windows 2000 and Windows .NET 20, 39, 103
 - disability 405
 - disk requirements
 - client 72
 - DB2 ESE 36
 - DB2 Personal Edition 19, 102
 - satellite control server 5
 - server 72
- ## E
- execution starting point, setting for satellite 267
 - execution targets
 - cataloging on Control Center 231
 - cataloging with Control Center 229
 - creating 256
 - description 188
 - managing password change 226
 - overview xx

- execution targets (*continued*)
 - target alias must match on satellite and Control Center 232
- exporting
 - client profile from Control Center 233
 - profiles 142
- F**
- fast communications manager (FCM)
 - enabling communications
 - between database partition servers 89
 - verifying port range availability 44
- fix batches
 - assigning to satellite 340
 - creating using model office 275
 - debugging 342
 - overview 194
 - purpose 219
 - querying results of with different fix batch 340
- fixed disks
 - hardware requirements 72
- FixPak
 - applying 25, 48, 61, 92, 108
- forward recovery
 - model office 304
 - test satellites 304
- G**
- group batches
 - application version 196
 - associated with application version 196
 - batch step components 188
 - bypassed during synchronization
 - if not available 197
 - cleanup 197
 - creating 197
 - creating test level from
 - production 215
 - debugging 265
 - example test-production cycle 204
 - level of application version 206
 - obsoleting production level 216
 - order of execution by
 - satellite 197
 - overview 194
 - parameterizing script 193
 - promoting 267
 - promoting test batch steps 201
- group batches (*continued*)
 - promoting test level to
 - production 214
 - setting execution starting point to
 - next batch step 182
 - setup 197
 - state and relationship with batch step 217
 - states of application-version
 - level 208
 - test-production cycle 199
 - testing
 - changing batch step 260
 - checking result 264
 - creating application version 257
 - creating authentication credential 255
 - creating level 0 258
 - creating target 256
 - editing level 0 259
 - enabling satellites 262
 - fixing problems 265
 - overview 254, 262
 - promoting to production 267
 - synchronizing test satellite 263
 - testing with model office 273
 - update 197
 - update batch in test level 213
- groups
 - administering xxiii
 - application version for each
 - version of end-user application 196
 - creating for synchronization
 - test 247
 - finding group-level
 - authentication credential 225
 - naming rules 379
 - overview xvi
 - scalable administration xxiii
- H**
- hardware requirements
 - fixed disk 72
- I**
- identifying and fixing failed satellite 337
- identifying failed satellite 337
- importing
 - client profile to model office 233
 - profiles 142
- installing
 - client 72
 - server 72
- installing model office 271
- installing new version of application
 - creating new application version
 - for group 322
 - creating system for test
 - deployment 323
 - deploying new application to
 - production satellites 325
 - determining application version
 - of satellite 325
 - overview 320
 - setting new application version
 - on satellite 321
- instance owning database partition
 - server
 - installing on Windows 40
- instances
 - migrating
 - on UNIX 165
- interactive installation
 - killing DB2 processes 125
- interface languages
 - changing
 - on UNIX 381
 - on Windows 383
- invoking DB2 installation program
 - from your installation application
 - for mass installation 314
- L**
- levels
 - application versions
 - states 208
- levels of application version
 - description 206
 - purpose 206
- life cycle of application version 201
- logs
 - determining synchronization
 - errors when log details are truncated 347
 - error return codes 345
 - on satellite 346
 - viewing for failed satellite 339
- M**
- mass copy
 - application data
 - considerations 319
 - DB2 considerations 318
 - operating system
 - considerations 319

- mass copy (*continued*)
 - overview 318
 - testing method 309
 - when to use 309
- mass deployment
 - completing 320
 - mass copy 318
 - mass installation 310
 - overview 309
 - testing method 309
- mass installation
 - available tools 310
 - completing satellite setup 317
 - copying DB2 installation files to media 314
 - copying response file and client profile files to media 314
 - customizing response file 313
 - customizing satellites 316
 - generating response file from model office 312
 - invoking DB2 installation program from your installation application 314
 - preparing distribution media 314
 - testing method 309
 - when to use 309
- migrating
 - databases 151, 153, 162, 167
 - DB2 153, 167
 - recommendations 146, 158
 - restrictions 144, 156
 - space considerations 148, 161
 - DB2 Administration Server (DAS) 167
 - instances, on UNIX 165
 - satellite control server
 - UNIX 155
 - Windows 143
- migration
 - planning 175
 - pre-Version 8 175
 - rebinding packages 175
- model office
 - cataloging execution targets on procedure 232
 - cataloging execution targets using client profile 233
 - cataloging local instance 234
 - cataloging on Control Center 231
 - characteristics 270
 - characteristics shared with satellite 269

- model office (*continued*)
 - completing setup of 235
 - database recovery 298
 - exporting client profile with response file generator 237
 - forward recovery 304
 - generating response file from 312
 - installing 271
 - overview xviii
 - post-deployment phase 275
 - production-deployment phase 274
 - purpose 269
 - recovering during development phase 299
 - refresh recovery 299
 - satellite control server and satellite control database must be cataloged on for synchronization 232
 - setting up 271
 - synchronizing 273
 - using
 - development and acceptance-testing phase 269
 - using for problem determination 275
 - version recovery 301

N

- naming conventions
 - general 373
- naming rules
 - for DB2 objects 375
 - for users, userIDs and groups 379
 - for workstations 377
- network
 - distributing the DB2 installation package 138
- Network File System (NFS) 75

O

- obsolete batches
 - batch steps in 217
- obsoleting production level 216
- obtaining information about a satellite failure 339
- online
 - documentation 27, 50, 109
 - help, accessing 394
- ordering DB2 books 394

P

- parameterized scripts 193
- partition database server
 - installing 49
 - verifying the installation, on Windows 49
- partitioned database environments
 - server installation, verifying on UNIX 93
 - server memory requirements on Windows 35
- password encryption, purpose 221
- passwords
 - changing for access to satellite control server 225
 - managing change of 224
- port numbers
 - range
 - verifying availability 44
- post deployment phase, model office considerations 275
- post-installation steps
 - satellite control server
 - AIX 7
 - Windows 9
- preparing distribution media for mass installation 314
- preparing for test
 - synchronization 240
- printed books, ordering 394
- problem determination
 - assigning fix batch to satellite 340
 - DB2 trace facility 344
 - db2sync -t command 329
 - debugging fix batch 342
 - determining synchronization errors when log details are truncated 347
 - error return codes 345
 - identifying and fixing failed satellite, overview 337
 - identifying failed satellite 337
 - installation overview 327
 - model office used for creating fix batch 275
 - obtaining information about a failure 339
 - overview 327
 - recreating satadmin.aut file 346
 - returning repaired satellite to production 343
 - satellite control server installation 328
 - satellite installation 328

- problem determination (*continued*)
 - satellite receives SQL0818 336
 - satellite software version 345
 - synchronization
 - application version not at satellite control server 334
 - application version not set at satellite 335
 - authentication error 335
 - overview 332
 - satellite can only synchronize once 332
 - satellite in failed state 333
 - satellite not enabled 333
 - synchronization configuration problems 329
 - synchronization test
 - authentication credentials missing from satellite 331
 - satellite control database not cataloged on satellite 330
 - satellite does not exist at satellite control database 331
 - satellite ID not set on satellite 329
 - version of DB2 on satellite and satellite control server not compatible 332
 - updating satadmin.aut file 346
 - production batches
 - batch steps in 217
 - production satellites
 - cataloging instances and databases on using client profile 237
 - overview xvii
 - recovery 306
 - production-deployment phase, model office considerations 274
 - profile
 - exporting 142
 - importing 142
 - promoting test level to production 214
- R**
- recovering control information in satellite environment 294
 - recovery
 - control information in satellite environment 294
 - satellite control database 296
 - satellite control server 296
 - satellite environment 293
 - recreating satadmin.aut file 346
 - refresh recovery
 - model office 299
 - test satellite 299
 - registry variables
 - DB2SATELLITEID 250
 - remote
 - commands, enabling on UNIX 90
 - remote access
 - configuring to a server database 140
 - Remote Command Service, setting user ID and password using response file 313
 - response file generator
 - customizing response file 313
 - exporting client profile 237
 - generating response file from model office 312
 - mass installation 310
 - response file keywords 121
 - COMP 121
 - DB2.AUTOSTART 121
 - DB2.DB2SATELLITEAPPVER 121, 122
 - DB2.DB2SATELLITEID 121
 - DB2SYSTEM 121, 122
 - FILE 121, 122
 - PROD 121, 122
 - REBOOT 121, 122
 - TYPE 121, 122
 - response files
 - considerations 114
 - creating
 - UNIX 128
 - Windows 132
 - generator
 - db2rspgn 124
 - overview 123
 - installation
 - configuring db2cli.ini 142
 - database partition servers 86
 - killing DB2 processes 126
 - making DB2 files available for 130
 - types 113
 - UNIX 126, 129
 - Windows 129
 - keywords
 - for Windows 122
 - for Windows 32-bit operating systems 116
 - running setup 133
 - samples 115
 - retrieving application version on satellite 283
 - returning repaired satellite to production 343
 - reverse engineer, definition 274
- S**
- satadmin.aut file
 - creation of on satellite 223
 - recreating 346
 - setting up satellite for synchronization test 249
 - updating 346
 - SATCTLDB database
 - considerations 6
 - creating 8
 - defaults 6
 - overview xv
 - satctldb.ddl 6
 - satellite
 - building applications 278
 - Satellite Administration Center
 - assigning fix batch to satellite 340
 - cataloging requirements for problem determination 229
 - debugging fix batch 342
 - identifying failed satellite 337
 - obtaining information about failure 339
 - overview xxiv
 - returning repaired satellite back to production 343
 - satellite control database
 - application version not defined at 334
 - availability 296
 - cataloging for application development 279
 - cataloging with Control Center 231
 - granting access to 243
 - granting privileges on bind files and stored procedures 247
 - must be cataloged in Control Center 229
 - not cataloged on satellite 330
 - overview xv
 - rebinding db2satcs stored procedure 336
 - recommended privileges 245
 - recovery 296
 - tables, overview 353

- satellite control server
 - application version not defined
 - at 334
 - availability 296
 - cataloging with Control Center 231
 - creating authentication credential for synchronization test 242
 - creating group for synchronization test 247
 - creating test satellite for synchronization test 248
 - disk requirements 5
 - granting access to satellite control database 243
 - installing
 - error messages 328
 - installing DB2 Enterprise Server Edition
 - overview 15
 - managing password change 225
 - migrating
 - UNIX 155
 - Windows 143
 - must be cataloged in Control Center 229
 - overview xiv
 - post-installation steps on AIX 7
 - post-installation steps on Windows 9
 - software requirements 5
 - storage of authentication credential 222
- satellite environment
 - synchronizing application
 - overview 277
- satellite environment
 - adding existing DB2 servers as satellites
 - overview 176
 - adding existing DB2 systems to environment
 - using scripts 178
 - authentication 221
 - building and running
 - synchronizing application 289
 - building applications 278
 - cataloging model office on Control Center 231
 - db2GetSyncSession API 283
 - db2QuerySatelliteProgress API 287
 - db2SetSyncSession API 282
 - db2SyncSatellite API 287
 - db2SyncSatelliteStop API 288
- satellite environment (*continued*)
 - db2SyncSatelliteTest API 284
 - example setup xxv
 - function of subgroup when staging a deployment xxv
 - general administration tables 357
 - groups
 - overview xvi
 - installing
 - preparation 3
 - model office
 - overview xviii
 - overview xiii
 - recovering Control Center directories 295
 - recovering control information 294
 - recovery 293
 - Satellite Administration Center xxiv
 - satellite control server
 - overview xiv
 - scalable administration xxiii
 - setting up
 - creating group batches 254
 - overview 239
 - preparing for test synchronization 240
 - testing synchronization capability of satellite 253
 - satellite synchronization
 - batch bypassed if not available 197
 - order of batch execution 197
 - satellite synchronization component
 - must be installed for application version to be set 249
- satellites
 - adding existing DB2 servers to satellite environment 176
 - adding existing DB2 systems to environment
 - using fix batch 179
 - adding new DB2 systems to environment
 - using fix batch 179
 - administering by group xxiii
 - application version 196
 - application version recorded on first synchronization 263
 - assigning fix batch to 340
 - authentication credentials not available 331
- satellites (*continued*)
 - authentication with target server 224
 - can only synchronize once 332
 - cataloging requirements when using as a client 232
 - changing password at satellite control server 225
 - changing password at target server 226
 - completing setup during mass installation 317
 - creating satadmin.aut file 223
 - customizing DB2 during mass copy 318
 - customizing during mass installation 316
 - debugging fix batch 342
 - disabled when in failed state 340, 341
 - execution of batch steps 188
 - execution starting point, setting 267
 - failed state, fixing 337
 - granting access to satellite control database 243
 - identifying failed 337
 - identifying information about failure 339
 - installing
 - error messages 328
 - installing DB2 Enterprise Server Edition
 - overview 15
 - installing DB2 Workgroup Edition
 - overview 15
 - interrupted synchronization session can result in authentication error 226
 - managing backup images 306
 - not created in satellite control database 331
 - order of batch execution 197
 - overview xvii
 - parameterizing script for 193
 - preparing for test synchronization 240
 - progress file 346
 - querying results of fix batch 340
 - recommended privileges
 - bind files and stored procedures 247
 - satellite control database 245
 - recreating satadmin.aut file 346

- satellites (*continued*)
 - returning back to production
 - after repaired 343
 - running db2trc on 344
 - Satellite Administration Center xxiv
 - satellite control database not cataloged 330
 - satellite control server and satellite control database must be cataloged on for synchronization 232
 - satellite in failed state 333
 - satellite not enabled 333
 - software version 345
 - storage of authentication credentials 223
 - storage of script during synchronization session 218
 - synchronization overview xxi
 - synchronizing using DB2 Synchronizer application 290
 - unique characteristics of DB2 Satellite Edition satellites 351
 - updating satadmin.aut file 346
 - version of DB2 incompatible with satellite control server 332
- scalar value, specifying 193
- scripts
 - authentication with target server 224
 - description 188
 - overview xx
 - parameterized 193
 - purpose 187
 - where stored on satellite during synchronization session 218
- servers
 - configuring remote access 140
 - memory requirements on UNIX 56
- set db2instance command 9
- setting application version on satellite 282
- setting the execution starting point for satellite 267
- setting up model office 271
- setting up satellite environment 239
- setup batches 197
- SMS
 - importing DB2 install file 136
 - installing DB2 products 135
- SMS packages, creating on the SMS server 137
- SMS servers, creating the SMS package 137
- software requirements
 - satellite control server 5
- software version 345
- SQLCODE
 - 1403 335
 - 3931W 331
 - 3932W 334
 - 3933W 332
 - 3934W 333
 - 3935W 333
 - 3950 332
 - 3951N 329
 - 3955 330
 - 3956N 335
 - 3966 331, 335
- states of an application version 213
- stored procedures
 - granting privileges on 247
- success code sets
 - description 188
 - overview xx
- synchronization
 - API overview 285
 - application version 196
 - application version does not exist at satellite control database 334
 - application version not set at satellite 335
 - application version of satellite recorded in satellite control database 263
 - authentication credentials 221
 - authentication with target server 224
 - checking results of 264
 - creating satadmin.aut file on satellite 223
 - db2GetSyncSession API 283
 - db2QuerySatelliteProgress API 287
 - db2SetSyncSession API 282
 - db2SyncSatellite API 287
 - db2SyncSatelliteStop API 288
 - db2SyncSatelliteTest API 284
 - finding group-level authentication credential 225
 - interrupted session can result in authentication error 226
 - managing password change to satellite control server 225
 - managing password change to target server 226
- synchronization (*continued*)
 - model office 273
 - overview xxi
 - preparing for test
 - creating authentication credential 242
 - creating group 247
 - creating test satellite 248
 - DB2SATELLITEID registry variable 250
 - granting access to satellite control database 243
 - granting administrative access to satellite control database 243
 - granting help desk access to satellite control database 243
 - granting privileges on bind files and stored procedures 247
 - granting satellites access to satellite control database 243
 - overview 240
 - recommended privileges 245
 - recommended table privileges 245
 - satadmin.aut file 249
 - satellite application version 249
 - testing satellite readiness 253
 - verifying the setup 252
- satellite can only synchronize once 332
- satellite control server and satellite control database must be cataloged on satellite 232
- satellite in failed state 333
- satellite not enabled 333
- setting authentication credentials using response file 313
- storage of script on satellite during session 218
- storage of scripts on satellite 218
- test mode
 - authentication credentials not available 331
 - db2sync -t command 329
 - satellite control database not cataloged 330
 - satellite does not exist at satellite control database 331

- synchronization (*continued*)
 - test mode (*continued*)
 - satellite ID not set on satellite 329
 - version of DB2 on satellite and satellite control server not compatible 332
- system configuration settings
 - recording before migrating DB2 149, 162

T

- table parameters 193
- tables
 - satellite administration 357
 - satellite control database, overview 353
- target servers
 - authentication of script 224
- test batch steps
 - execution by test satellites 200
 - promoting to production 201
- test batches
 - batch steps in 217
- test levels
 - update batch 213
- test satellites
 - application version for test synchronization 249
 - cataloging instances and databases on
 - overview 236
 - using client profile 237
 - checking results of synchronization 264
 - creating for synchronization test 248
 - database recovery 298
 - enabling to execute test-level batches 262
 - execution of production batch steps 217
 - execution of test batch steps 200
 - forward recovery 304
 - overview xvii
 - refresh recovery 299
 - satadmin.aut file for test synchronization 249
 - setting DB2SATELLITEID for test synchronization 250
 - synchronizing to execute test-level batches 263
 - verifying setup of 252
 - version recovery 301

- testing
 - group batches 262
- troubleshooting
 - DB2 documentation search 402
 - online information 404
- tutorials 406

U

- unassigned batches
 - overview 194
- UNIX
 - changing the DB2 interface language 381
 - creating a response file 128
 - disk requirements for DB2 ESE/WSE 57
 - enabling Control Center administration 91
 - enabling remote commands 90
 - installing a database partition server using the DB2 Setup wizard 81
 - installing a partitioned DB2 server 64
 - installing database partition servers using a response file 86
 - installing DB2 online documentation 62, 94
 - installing DB2 servers
 - UNIX 52
 - using the DB2 Setup wizard 58
 - memory requirements
 - partitioned DB2 servers 71
 - servers 56
 - migrating instances 165
 - response file installation 126, 129
 - updating the node configuration file 87
 - verifying a partitioned database server installation 93
- update batches 197
 - test level 199, 213
- updating satadmin.aut file 346
- user accounts
 - required for installation of DB2 servers on Windows 21
- user IDs
 - naming rules 379
- users
 - creating for a partitioned DB2 server
 - AIX 78

- utilities
 - response file generator 123

V

- version recovery
 - model office 301
 - test satellites 301

W

- Windows
 - changing the DB2 interface language 383
 - creating response files 132
 - installation requirements
 - DB2 servers 17
 - partitioned DB2 servers 33
 - installing
 - database partition servers 45
 - DB2 Personal Edition 97, 98, 100
 - DB2 server 14
 - DB2 servers 12
 - DB2 servers with the DB2 Setup wizard 23
 - DB2 servers, user accounts 21
 - instance owning database partition servers 40
 - online documentation 27, 50, 109
 - partitioned DB2 server 29
 - partitioned DB2 servers 30
- memory requirements
 - DB2 Personal Edition 102
 - partitioned DB2 servers 35
- migrating databases on DB2 Personal Edition 173
- migrating DB2 Personal Edition 169, 170
- preparing to migrate DB2 Personal Edition 170
- Response file installation 129
- response file keywords 116
- running setup from the client workstation 133
- server memory requirements 19
- setting a shared access directory 131
- starting the DB2 Setup wizard 105
- verifying
 - partition database server installation 49

- Windows .NET, extending the directory schema 20, 39, 103

Windows 2000
 extending the directory
 schema 20, 39, 103
workstations
 (nname), naming rules 377

Contacting IBM

In the United States, call one of the following numbers to contact IBM:

- 1-800-237-5511 for customer service
- 1-888-426-4343 to learn about available service options
- 1-800-IBM-4YOU (426-4968) for DB2 marketing and sales

In Canada, call one of the following numbers to contact IBM:

- 1-800-IBM-SERV (1-800-426-7378) for customer service
- 1-800-465-9600 to learn about available service options
- 1-800-IBM-4YOU (1-800-426-4968) for DB2 marketing and sales

To locate an IBM office in your country or region, check IBM's Directory of Worldwide Contacts on the web at www.ibm.com/planetwide

Product information

Information regarding DB2 Universal Database products is available by telephone or by the World Wide Web at www.ibm.com/software/data/db2/udb

This site contains the latest information on the technical library, ordering books, client downloads, newsgroups, FixPaks, news, and links to web resources.

If you live in the U.S.A., then you can call one of the following numbers:

- 1-800-IBM-CALL (1-800-426-2255) to order products or to obtain general information.
- 1-800-879-2755 to order publications.

For information on how to contact IBM outside of the United States, go to the IBM Worldwide page at www.ibm.com/planetwide



Printed in U.S.A.

GC09-4823-00

