Installing ObjectStore for Windows

Release 6.2
Installing ObjectStore for Windows

ObjectStore, Release 6.2 for all platforms, November 2004

© 2004 Progress Software Corporation. All rights reserved.

Progress® software products are copyrighted and all rights are reserved by Progress Software Corporation. This manual is also copyrighted and all rights are reserved. This manual may not, in whole or in part, be copied, photocopied, translated, or reduced to any electronic medium or machine-readable form without prior consent, in writing, from Progress Software Corporation.

The information in this manual is subject to change without notice, and Progress Software Corporation assumes no responsibility for any errors that may appear in this document.

The references in this manual to specific platforms supported are subject to change.

A [Stylized], Allegrix, Allegrix & Design, Business Empowerment, eXcelon, ObjectStore, Progress, Powered by Progress, Empowerment Center, Progress Empowerment Center, Progress Empowerment Program, Progress Fast Track, Progress Profiles, Partners in Progress, Partners en Progress, Progress en Partners, Progress in Progress, P.I.P., Progress Results, Progress Software Developers Network, ProVision, ProCare, ProtoSpeed, SmartBeans, SpeedScript, Technical Empowerment, and WebSpeed are registered trademarks of Progress Software Corporation or one of its subsidiaries or affiliates in the U.S. and/or other countries. AccelEvent, A Data Center of Your Very Own, AppsAlive, AppServer, ASPen, ASP-in-a-Box, BusinessEdge, Cache-Forward, Empowerment Center, Fathom, Future Proof, IntelliStream, ObjectCache, ObjectStore Event Engine, ObjectStore RFID Accelerator, OpenEdge, PeerDirect, POSSE, POSSENET, ProDataSet, Progress Business Empowerment, Progress for Partners, Progress OpenEdge, PSE Pro, PS Select, SectorAlliance, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, Trading Accelerator, WebClient, and Who Makes Progress are trademarks or service marks of Progress Software Corporation or one of its subsidiaries or affiliates in the U.S. and other countries.

Any other trademarks or service marks contained herein are the property of their respective owners.

ObjectStore includes software developed by the Apache Software Foundation (http://www.apache.org/). Copyright © 2000-2003 The Apache Software Foundation. All rights reserved. The names “Ant,” “Xerces,” and “Apache Software Foundation” must not be used to endorse or promote products derived from the Products without prior written permission. Any product derived from the Products may not be called “Apache”, nor may “Apache” appear in their name, without prior written permission. For written permission, please contact apache@apache.org.

September 2004
# Contents

## Chapter 1

**Overview** ......................................................... 3  
ObjectStore Components ........................................... 3  
ObjectStore DBMS .................................................. 4  
C++ Interface ...................................................... 4  
Java Interface ...................................................... 4  
Java Middle Tier Library (JMTL) ................................. 5  
Dynamic Data Modeling Language ............................... 5  
ObjectStore Platform Configurations ............................ 5  
Types of Installation Procedures ................................. 6  
New installation .................................................... 6  
Upgrade from Release 6.0 or 6.1 ................................. 6  
Upgrade from Release 5.1 .......................................... 8  
Upgrading from 6.2 to a Later Service Pack .................... 8  
Adding a Component ................................................ 8  
Configuration ....................................................... 8  
Installation Concepts ............................................... 9  
Development Configuration ....................................... 13  
Run-time Configuration .......................................... 13  
Types of Databases ............................................... 13  
What is the Transaction Log? .................................... 14  
What is a rawfs? .................................................. 14  
Purchasing Technical Support .................................... 15

## Chapter 2

**System Requirements for ObjectStore** ........................... 17  
Supported Platforms ............................................... 17  
Hardware Requirements .......................................... 17  
Memory ............................................................ 17  
Disk Space ........................................................ 18  
Software Requirements ........................................... 19  
Installation Media for ObjectStore .............................. 19  
Obtaining Installation Media ...................................... 19
Contents

Description of ObjectStore Releases ........................................... 20

Chapter 3 Installing and Configuring ObjectStore ......................... 23
Are You Upgrading? ................................................................. 23
Preparing for Installation ......................................................... 24
Installing ObjectStore ............................................................. 24
  Typical Installation ............................................................... 25
  Custom Installation .............................................................. 26
ObjectStore Environment Variables ............................................. 29
Verify Installation ................................................................. 30
  Verify Installation Directory Contents ...................................... 31
  Verify ObjectStore Services .................................................. 32
  Verify a C++ Development Client .......................................... 33
  Verify a C++ Run-time Client ................................................. 33
  Verify a Java Client .............................................................. 34
Configuration Tasks ............................................................... 34
Upgrading from ObjectStore 6.0 or 6.1 ........................................ 36
Upgrading from Release 5.1 ...................................................... 37

Chapter 4 Miscellaneous Information .......................................... 39
Uninstalling ObjectStore ......................................................... 39
  Adding ObjectStore Components .......................................... 40
  Resolving Client-Only Schema Database Issues ........................ 40
Troubleshooting Installation Problems ....................................... 41
Chapter 1
Overview

ObjectStore is an Object Database Management System (ODBMS) that enables applications to store and manage any type of data, no matter what the complexity. Its distributed process architecture allows applications to scale and enables in-memory access to data of any type—C++, Java, XML. The ObjectStore ODBMS supports any variety of distributed processing through its flexible process architecture and ensures transactional consistency in a wide variety of network configurations. The ObjectStore server process supports clients on many different platforms; see the Support Matrix on the Technical Support web site (http://www.objectstore.net/support/matrix) for current information.

ObjectStore applications running on different platforms can all access the same data. ObjectStore has all of the necessary features for system management, including on-line backup and restore, archive logging and recovery, replication, transparent failover and many other operations necessary in an enterprise ODBMS.

ObjectStore Components

ObjectStore contains several components that can be used in a development or in a deployment environment. This document refers to the latter as a run-time environment. The type of ObjectStore installation depends on the type of environment required. Here are the major components of an ObjectStore environment.
ObjectStore Components

ObjectStore DBMS

The ObjectStore DBMS is the database server and associated files that make up the core ObjectStore services that can serve both C++ and Java applications. This document refers to the DBMS as the server because in an ObjectStore application, the ObjectStore server communicates with any type of application (C++ or Java) and accesses ObjectStore databases on behalf of the C++ or Java client applications. Along with the ObjectStore server, the DBMS also includes a number of utilities and libraries that can be used to administer databases in an ObjectStore environment. Which utilities and libraries are installed will depend upon whether the environment is for C++ or Java.

C++ Interface

C++ developers write client applications using the ObjectStore C++ Application Programming Interface (API) to access ObjectStore databases in a client process that communicates with the ObjectStore server process. Developers access data in their databases as they would any C++ object in program memory, but use the ObjectStore C++ API for controlling placement of their objects in the database and for transactional control. ObjectStore includes a collections library that provides application developers the ability to aggregate, query, and index C++ objects in a database. ObjectStore supports C++ applications written on different platforms and compilers, all accessing the same data in a distributed environment. The C++ API also includes the C++ Middle-tier Library (CMTL) which enables developers to implement transactionally-consistent distributed caches.

Java Interface

Java developers write client applications using the ObjectStore Java API to access ObjectStore databases in a Java Virtual Machine (JVM) which communicates with the ObjectStore server process. Developers access data in their databases as they would any Java object in program memory, but use the ObjectStore Java API for controlling placement of their objects in the database and for transactional control. The JDK Collections and ObjectStore Collections classes provide application developers the ability to aggregate, query, and index Java objects in a database. ObjectStore supports Java applications written on different platforms, all accessing the same data in a distributed environment.
Chapter 1: Overview

Java Middle Tier Library (JMTL)

The Java Middle Tier Library (JMTL) integrates with several application servers for either EJB or J2EE JTA transaction support (or both). Developers use JMTL to write middle-tier applications that utilize the power of transactionally-consistent distributed data caches. The JMTL component utilizes the Java interface to ObjectStore, along with a specialized set of classes which enable the batching and routing of transactional activity to be sent to the ObjectStore server process.

Dynamic Data Modeling Language

The Dynamic Data Modeling Language (DDML) provides an easy-to-use model for the creation of name/value pairs for flexible schema data access in an ODBMS. With the C++ interface, developers can use DDML to extend schemas at run-time.

ObjectStore Platform Configurations

ObjectStore is a product that may be installed on multiple types of hardware, operating systems, and in conjunction with a variety of compilers, language-related libraries, JDKs and JREs. The product is built to run on a set of platforms.

ObjectStore supports a number of platforms including Windows, Solaris, and Linux. For each platform, there are one or more compilers that ObjectStore is built with in order support applications built with those compilers. In this document, the combination of compiler/operating system is referred to as a platform configuration. In order to install ObjectStore, it is necessary to understand what platform configuration you require for your application.

For a list of platform configurations supported by this release of ObjectStore, see the Support Matrix (www.objectstore.net/support/matrix) on the Technical Support web site. This information is also available in the Release Notes.
Types of Installation Procedures

The procedures that you follow in order to install this release of ObjectStore or any of its components will depend on what your goal is. Here is a summary of the ways in which the ObjectStore installation procedures may be used on a given platform:

- New installation on page 6
- Upgrade from Release 6.0 or 6.1 on page 6
- Upgrade from Release 5.1 on page 8
- Upgrading from 6.2 to a Later Service Pack on page 8
- Adding a Component on page 8

New installation

This is the most straightforward type of installation because there is no existing application or set of databases that need to be migrated. These procedures should be followed when ObjectStore is being installed for the first time for new development or for setting up a system for deployment.

Upgrade from Release 6.0 or 6.1

Upgrading from ObjectStore 6.0 or 6.1 is straightforward, unless your application requires a different compiler for 6.2 from the one used in the earlier release. You should first make sure that the platform configuration is supported for this release and service pack. For a list of supported platforms, see the Support Matrix (www.objectstore.net/support/matrix) on the Technical Support web site.

For an upgrade from 6.0 or 6.1 to 6.2, application developers need to re-compile and re-link their applications. The rules concerning compatibility of ObjectStore processes in a mixed-release environment are as follows:

- ObjectStore 6.2 servers can fulfill requests from 6.0.x, 6.1.x, and 6.2 clients.
- ObjectStore 6.0.x and 6.1.x servers cannot fulfill requests from 6.2 clients.
- ObjectStore 6.2 clients can use 6.0.x and 6.1.x databases that are managed by 6.2 servers if the schema of the 6.2 client application is compatible with the schema of the code that generated the 6.0.x or 6.1.x database. The schemas are compatible if at least one of the following is true:
  - The client application and the code that generated the database were compiled by the same compiler version.
Chapter 1: Overview

- When the code that generated the database was compiled, it was neutralized for the machine on which the client application was compiled.

This means that if you are upgrading a number of host systems in a distributed environment, your first step must first upgrade the host (or hosts) where the ObjectStore server process resides.

In general, databases created in 6.0 or 6.1 will be compatible for 6.2 applications. The exceptions to this will be if there is a major compiler change (see below).

An upgrade requires that you first ensure that all applications running on a given host that use the ObjectStore server process are shut down, the ObjectStore server process has checkpointed the databases it has open (propagated the transaction log to the physical databases), and that the necessary ObjectStore processes have been cleanly shut down. The checkpointing and shutting down of the ObjectStore server process is only necessary if you are upgrading the host where the ObjectStore server process is running.

When 6.2 is installed, the installation process will ask if you want to overwrite the existing ObjectStore binaries, delete and install them, or install them in a new location. It will be the responsibility of the user doing the installation to make sure that all databases are properly backed up prior to the upgrade.

In addition to database compatibility between releases, there are some code changes that you need to make if your applications use obsolete or deprecated APIs. Obsolete and deprecated APIs are detailed in the Migration Guide available on the Technical Support Web site at www.objectstore.net/documentation/migration.

ObjectStore does not automatically support an upgrade of an application whereby the compiler is changing. Please refer to the Migration Guide for instructions for how to upgrade to a different compiler. A common compiler change as customers upgrade to this release is Visual C++ 6 to Visual Studio .NET on Windows.

As more platforms/compilers are available for ObjectStore 6.2, more options will be available for developers to upgrade their ObjectStore environments.

For more information on upgrading Release 6.0 or 6.1 databases and applications, see Upgrading from ObjectStore 6.0 or 6.1 on page 36.
Upgrade from Release 5.1

If you have an existing ObjectStore Release 5.1 installation and want to upgrade to Release 6.2, you need to perform a database migration and modify your application source code in order to use your databases and applications with Release 6.2. Consult the Support Matrix (www.objectstore.net/support/matrix) on the Technical Support web site which lists the platform configurations supported for this release of ObjectStore. Some platforms supported in Release 5.1 are not supported in Release 6.2.

For more information on upgrading Release 5.1 databases and applications, see Upgrading from Release 5.1 on page 37.

Upgrading from 6.2 to a Later Service Pack

This procedure is very simple because it involves the installation of the latest release of ObjectStore over your existing release. It does not involve any application source code changes or database migrations. It is generally drop-in or link compatible, depending on the nature of the service pack.

As with any ObjectStore upgrade, you must ensure that all databases have been backed up, all ObjectStore applications have been shut down and all ObjectStore services are no longer running. This type of upgrade should be done to overwrite the existing ObjectStore installation.

Adding a Component

If, during installation, you choose not to install the entire ObjectStore product suite, you can use the ObjectStore installation program to add additional components at a later date.

An example of this would be if you installed the Java Interface and now want to install the JMTL component. For more information, see Adding ObjectStore Components on page 40.

Configuration

On Windows, you can use the installation process to do some configuration tasks such as initializing the transaction log file or setting up ObjectStore server process parameters.

See Configuration Tasks on page 34 for more details.
Installation Concepts

The main processes in an ObjectStore environment depend upon the components installed and whether the application is a Java application or a C++ application. The core ObjectStore environment is C++. If your application is written in Java, it will execute some C++ code at the storage management level automatically. This is why Visual Studio C++ is required in an ObjectStore environment.

All access to ObjectStore databases is done via the ObjectStore server process (osserver). The ObjectStore server process must reside on the same host where the physical databases reside. This eliminates the need for the ObjectStore server process to access databases over the network.

On the server side, the resources are the ObjectStore server process, its databases that it is accessing, and the ObjectStore transaction log file which also must reside on the same host as the ObjectStore server process. There is always one and only one ObjectStore transaction log file, no matter how many databases the ObjectStore server process is managing. A database can only be managed by a single ObjectStore server process. The ObjectStore transaction log is a special file that holds all transaction data that needs to be propagated to the databases that the ObjectStore server process is managing. The ObjectStore transaction log is a critical file that should never be removed unless you are sure that all data has been propagated to the databases.

Any number of ObjectStore clients can access databases via one or more ObjectStore server processes. (each ObjectStore server process needs to be running on a separate host). In order for an ObjectStore client to access a database, it communicates with two processes: the ObjectStore server process and the ObjectStore cache manager process (oscmgr6). The ObjectStore server process grants access to clients and the ObjectStore cache manager process keeps track of locks that all clients have on a particular host with any number of ObjectStore server processes they are communicating with which could be on multiple hosts. The ObjectStore client can either be a C++ process that is linked with the ObjectStore libraries or a JVM that utilizes the ObjectStore classes in the ObjectStore .jar files.

When JMTL is installed and configured, either for development or run time, the process architecture involves the use of an application server. There are many design choices that can be made in the use of JMTL and how it will utilize an application server. A typical configuration would be one where multiple JVMs are instantiated, each of which could be an ObjectStore client process.
Installation Concepts

The following figure illustrates a simple ObjectStore process architecture:

In this environment, there is a single ObjectStore server process and multiple ObjectStore client processes, all running on the same host. This environment requires a typical installation if you need both C++ and Java environments. If you need only a C++ environment, then perform a custom installation, installing the C++ interface and ObjectStore DBMS, and selecting both client and server.

The following figure illustrates a multi-host ObjectStore process architecture involving a single ObjectStore server process:

In this environment, there is a single ObjectStore server process and multiple ObjectStore client processes. The ObjectStore server process has been set up
on Host1 which is equipped with perhaps a larger set of disks for databases. The ObjectStore clients can run on either Host1 or Host2. This environment requires a typical installation if you want to install both C++ and Java environments on Host 1.

If you require only a C++ environment, then perform a custom installation deselecting the Java interface and installing both client and server on Host 1. For Host 2, perform a custom installation and deselect the Java interface, installing the client only.

The following figure illustrates a multi-host ObjectStore process architecture involving multiple ObjectStore server processes:

In this environment, there are multiple ObjectStore server processes running on multiple hosts and multiple ObjectStore client processes. The ObjectStore server process has been set up on Host1 which is equipped with perhaps a larger set of disks for databases. The ObjectStore clients can run on either Host1 or Host2.

This environment requires a typical installation if you want to install both C++ and Java environments on Host 1 or Host2. If you require only a C++ environment, then perform a custom installation, deselecting the Java interface and installing both client and server on Host 1 and Host2.
The following figure illustrates a very simple process architecture where JMTL is installed, along with an application server.

In this environment, there is a single ObjectStore server process and multiple ObjectStore client processes. Some clients are running on Host1 and on Host 2, there is an application server which is using JMTL. The ObjectStore clients can run on either Host1 or Host2.

This environment requires a typical installation if you want to install both C++ and Java environments on Host 1. If you require only a C++ environment, then you would perform a custom installation, but would deselect the Java interface and install both client and server on Host 1. For Host2, perform a custom installation where you select the Java Interface and JMTL, installing the client only.
Chapter 1: Overview

Development Configuration

A development configuration typically involves the installation of all ObjectStore components for development and run time along with the appropriate compilers or JDK. C++ developers need access to an ObjectStore server process in order to build the application. This is because the C++ build process involves the creation of an application schema database which must be created using the ObjectStore server process. A developer needs access to the C++ include files and libraries or the Java .jar files in order to build and test the application. Any number of developers can share the resources of an ObjectStore server process on a shared host, provided that the developers who need to access databases (during build or testing) have access to a host with an ObjectStore server process and the databases used by the application for building or testing will be local to that host where the ObjectStore server process is running.

Run-time Configuration

A run-time or deployment configuration will be one that should have a predetermined configuration, based upon the application needs. In many cases, it will involve a dedicated host that runs only the ObjectStore server process. All clients are on different hosts. It may involve a pool of hosts that will have ObjectStore server processes on them. The ObjectStore server process hosts will need to be capable of having very large disk storage capabilities. Memory requirements are not as important for the ObjectStore server process.

The ObjectStore client processes will require the most memory. Each client will need a minimum amount of memory and backing disk store for paging. How many clients one configures to run per host will depend on the application design.

You can run both the ObjectStore server process and one or more ObjectStore client processes on the same host. This is an application design choice.

Types of Databases

An ObjectStore database is a file in the operating system file system. The exception to this is a rawfs database which will be described in What is a rawfs? on page 14. The only ObjectStore process which physically accesses the database is the ObjectStore server process. A database contains an application schema which describes the types of objects that this database can contain. Only instances of objects of the defined types may be stored in the database. An ObjectStore database can only be accessed or view by an
application or specialized tools and utilities that are developed to access the
data in the database. An ObjectStore database can only be served by a single
ObjectStore server process. An application can access any number of
ObjectStore databases at the same time. The physical database files must
reside on the local file system of the host where the ObjectStore server
process is running. A special type database with the suffix of .adb and .ldb
is used by applications in order to ensure that the schema of objects being
used by the client applications match the schema of the database containing
the data. The special databases are called application schema databases and
library schema databases. Since these special files are databases, they can
only be accessed by the ObjectStore server process. These databases must be
placed on a host file system where an ObjectStore server process is running.

What is the Transaction Log?

The ObjectStore transaction log is a special file that is only accessed by the
ObjectStore server process. It must be located on the same file system where
the ObjectStore server process is running. In addition, for performance
reasons, it is recommended that the ObjectStore transaction log is configured
to be located on a different physical disk drive than the disk drive which will
hold the databases that the ObjectStore server process will access. The
ObjectStore transaction log is the file that contains the active transactions for
all databases that the ObjectStore server process is serving to clients. At pre-
configured intervals, the contents of the log file for committed transactions is
propagated to the databases. For this reason it is extremely important that
the ObjectStore transaction log not be removed unless a special configuration
is being made and the system administrator has ensured that all transaction
data has been forced to be propagated to the databases. This forcing of the
propagation can be achieved by checkpointing the ObjectStore server
process and by shutting down the ObjectStore server process. In order for
this effective checkpointing or shutdown to occur, it is also necessary that the
system administrator can ensure that no ObjectStore client applications
access the databases served by this ObjectStore server process. Once the
ObjectStore server process is completely shut down, the ObjectStore
transaction log can be safely removed. This enables the system administrator
to initialize a new ObjectStore transaction log file either in the same location
or in a different location.

What is a rawfs?

The term rawfs is the term for the ObjectStore RAW File System. This is a
special area or partition on a disk that is reserved for use only by the
Chapter 1: Overview

ObjectStore server process. Only one ObjectStore server process may access a rawfs. One advantage of a rawfs is that it can span multiple disk partitions, allowing for very large databases. Another advantage is that the files (databases) and directory structure that make up the rawfs are not visible from the operating system using normal commands (dir or ls). This special file system is only visible via the ObjectStore commands or by an ObjectStore client process.

A rawfs is expandable, but it cannot be made smaller. If you want a rawfs to be smaller, you have to back up the databases in the rawfs, remove the rawfs and recreate a smaller one.

If you are planning on using a rawfs partition, you should make sure the partition has been already created prior to the configuration of the partition to be used by ObjectStore. If you are planning on using a file for a rawfs, you should make sure that there is adequate disk space for the rawfs that you want it to contain. Finally, a rawfs for a host must be local to that host.

Purchasing Technical Support

To obtain information about purchasing technical support, contact your local sales office listed at www.objectstore.net/contact_us (worldwide) or call 1-781-280-4757 (in North America). When you purchase technical support, the following services are available to you:

- You can send e-mail questions to support@objectstore.net. Remember to include your serial number in the body of the message.
- You can call the Technical Support organization to get help resolving problems. If you are in North America, call 1-781-280-4005. If you are outside North America, refer to the Technical Support Web site, www.objectstore.net/techsupport.
- You can access resources available on the Technical Support Web site. These resources include:
  - A template for submitting support requests. This template can speed up response time helping you provide the necessary details.
  - The Knowledge Center, a database of technical information of interest to ObjectStore users. The Knowledge Center contains (among other things) solutions to technical problems encountered by users; see www.objectstore.net/techsupport/knowledge_center.
  - Online documentation for all ObjectStore products.
Purchasing Technical Support

- White papers and short articles about using ObjectStore products.
- Sample code and examples.
- The latest versions of ObjectStore products, service packs, and patches available for downloading.
- The Support Matrix, a list of platform configurations that are supported by this release; see www.objectstore.net/techsupport/support_matrix.
- Support policies.
- Local phone numbers and hours when support personnel can be reached.
Chapter 2
System Requirements for ObjectStore

Supported Platforms

The Support Matrix (www.objectstore.net/support/matrix) contains information about which platform configurations are supported for this release. A supported platform configuration means that it has been thoroughly tested prior to product release. A maintained platform configuration means that it has been known to work and has received some testing, but it has not been tested thoroughly. See the Technical Support site for more details concerning product support.

Hardware Requirements

Before you install the ObjectStore software, make sure that your machine environment meets the following requirements.

Memory

The amount of physical memory required will depend on the installation configuration required by the development or deployment environment. The ObjectStore server process does not require much physical memory, but a minimum of 128 MB is recommended. If you are running ObjectStore client processes on a host, you will need more physical memory. Most of the ObjectStore processing is done in the client. If the application is written to require large transactions, then more physical memory will be a benefit.
Hardware Requirements

There are some applications that are written with the expected performance of in-memory access speeds of very large datasets. These configurations require a great deal of physical memory.

Disk Space

Disk Space for the Application and ObjectStore

There are several considerations for estimating the amount of disk space required. One consideration is the types of files required in an ObjectStore environment:

- Application binaries and related files — all the files required to run the application (excluding ObjectStore databases and client caches) possibly on a shared device to be used by many clients
- ObjectStore installation files — disk space requirements depend on which components you install.
  
  - ObjectStore DBMS and C++ interface
  - Java interface
  - JMTL
  - DDML
  - Documentation

- ObjectStore client cache space — client cache requirements depend on the application. On Windows, the cache requirements (cache size * number of clients on that host) need to be added to the system swap space.
- ObjectStore server
- ObjectStore transaction log file — the maximum size will depend on the application and whether it has large transactions or is using Multi-version Concurrency Control (MVCC). Create this file on a physical disk that is local to the host running the ObjectStore server process and is a different physical disk than all of the databases that will be served by this ObjectStore server process.
- Application Databases — make sure that there is enough physical disk space for all databases the application may need (even for future growth). All databases served by this ObjectStore server process must be on a disk that is local to the host where the ObjectStore server process is running. Consider disk space for possible on-line backup, archive logging, replication if the application requires it

<table>
<thead>
<tr>
<th>Component</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectStore DBMS</td>
<td>105 MB</td>
</tr>
<tr>
<td>C++ interface</td>
<td></td>
</tr>
<tr>
<td>Java interface</td>
<td>17 MB</td>
</tr>
<tr>
<td>JMTL</td>
<td>9 MB</td>
</tr>
<tr>
<td>DDML</td>
<td>9 MB</td>
</tr>
<tr>
<td>Documentation</td>
<td>70 MB</td>
</tr>
</tbody>
</table>
Disk Space for Installation Processing
When installing ObjectStore, you also need to allocate temporary disk space for:

- Installation package — 303 MB (including debug libraries)
- Temporary space required to do the installation (unpackaging)

The amount of disk space for the above steps, will depend upon the components to be installed. You do not need additional disk space for the installation package if you are installing from CDROM. However, if you download the package from Technical Support, you’ll need as much as 303 MB of extra disk space, depending on which components you download.

Software Requirements
In order to install this release of ObjectStore, you need to make sure that the appropriate versions of operating systems, C++ compilers/libraries (if C++ client applications will be developed or run), JDKs or JREs, application servers (if JMTL is to be used with an application server), and the appropriate patches for the above software is installed. Information about specific versions and patch-levels for software that supports this release of ObjectStore is documented in the README.htm for this release. In addition, current information about supported and maintained configurations can be found in the Support Matrix (www.objectstore.net/support/matrix).

In addition to the platform configuration information for your installation, it is recommended that you read the product documentation using a Web browser (Netscape 4.7 or greater or Internet Explorer 5.0 or greater) to view the Webhelp or Adobe Acrobat Reader (4.0 or greater) to view the PDFs.

Installation Media for ObjectStore

Obtaining Installation Media
The files necessary for installing ObjectStore can be obtained either from a CDROM or from the Technical Support download area. A CDROM can be obtained by contacting your sales representative or Technical Support (www.objectstore.net/support. or 781-280-4005).
In order to gain access to the Technical Support FTP server, you must contact Customer Service (objectstore-customerservice@objectstore.com or 781-280-4005) to obtain the necessary username and password.

Each platform configuration for the installation of the ObjectStore components is in its own directory. In addition, on the download site, there will be a directory name which reflects the name of the release of ObjectStore. It is important to understand which platform configuration you need to install.

The CD for installing ObjectStore on a particular platform configuration contains the following files:

- README.htm: Important information about this release
- ReleaseNotes.pdf: New and changed features of this release
- InstallationGuideForWindows.pdf: Installation instructions
- setup.exe: Installation program
- *.cab: Installation contents

If you download the ObjectStore installation package, you get the following files:

- README.htm: Important information about this release
- InstallationGuideForWindows.pdf: Installation instructions
- ObjectStore.exe: Installation program

Note that when you run the ObjectStore.exe program, you get the files that are provided on the CD, including ReleaseNotes.pdf.

### Description of ObjectStore Releases

ObjectStore releases are given numbers followed by a Service Pack number. For example, ObjectStore 6.2 is the first release and subsequent releases will be named 6.2 Service Pack 1, 6.2 Service Pack 2, etc. Service Packs are *drop-in compatible* and it is expected that customers will upgrade to Service Packs as they are released. In addition to Service Packs, Technical Support sometimes releases *patches* or *quick drops*. These are very specific libraries that are intended to address a software problem that needs to be fixed before a full Service Pack can be tested and released. The installation of Service Packs are full installations or upgrades. The installation of patches or quick drops are
typically the replacement of a library. When a Service Pack is released, it contains all patches and quick drops that were distributed prior to the Service Pack.
Installation Media for ObjectStore
Chapter 3
Installing and Configuring ObjectStore

This chapter provides procedures for installing and configuring ObjectStore.

Are You Upgrading?

If you are upgrading ObjectStore, you must do the following before proceeding with the installation:

- If ObjectStore applications are running on the host you are upgrading and those applications use the ObjectStore server process, you must ensure that they are no longer running.
- If the server process is running on the host you are upgrading, you must do the following:
  - Cleanly shut down any ObjectStore processes running on the host.
  - Use the following command line to checkpoint the database that the ObjectStore server has open (checkpointing propagates information in the transaction log to the database):
    ```
    ossvrchkpt host_name
    ```
  - Shut down the ObjectStore server (osserver).

Note

If you want to install Release 6.2 of ObjectStore in a location where an existing ObjectStore installation currently resides, you must either move the existing installation to another location or uninstall it. If you are going to uninstall a previous version of the software, you need to decide on an upgrade procedure to minimize interruptions in service. See Upgrading...
Preparing for Installation

Review the Chapter 1, Overview, on page 3 before you begin the installation. Also, read the README file for any last-minute information concerning the installation process. Before you install, determine the answers to the following questions:

- What is the path of the destination directory for the ObjectStore installation?
- Which ObjectStore components do you want to install?
- Is there adequate disk space for installation?
- Will the host be used for developing ObjectStore applications?
- Will this host will be running an ObjectStore server process? If so, you should also determine the following:
  - Where will the transaction log will reside? Note that the transaction log must reside on the host where the ObjectStore server process is running.
  - Is the amount of disk space adequate for the databases and the transaction log?
- If you are doing a client-only installation, which host will be used for the ObjectStore server process for the clients? Will the clients have access to a file system local to that ObjectStore server process?
- Where will the physical databases reside?

Installing ObjectStore

The following sections give two different procedures for installing ObjectStore:

- Typical Installation on page 25 assumes that you want to install all of the ObjectStore components and that you want the installation program to do most of the configuring. The typical installation requires minimal involvement on the part of the user.
Chapter 3: Installing and Configuring ObjectStore

- Custom Installation on page 26 gives you greater control over the installation process and accordingly asks you to make more decisions about the process than does the typical installation.

Typical Installation

The following procedure is for users who want to install all of the ObjectStore components and who want to let the installation program do most of the configuring. Note that you can always reconfigure the installation at a later time, after you have installed ObjectStore.

**Note**

If you are upgrading, you must first uninstall the currently installed release of ObjectStore, as described in Uninstalling ObjectStore. The following procedure assumes that you do not have ObjectStore currently installed on the target machine:

1. Run the `setup` program that is included in the installation package.
2. The `setup` program displays the License Agreement and asks you to indicate your acceptance of the Agreement by clicking Yes. The program next displays the copyright notice and asks you to indicate your acknowledgement by clicking OK.
3. You are asked to supply user information. If `setup` can retrieve this information from the Windows registry, it will display it.
4. You are asked to enter the path of the destination directory where you want ObjectStore to be installed.
5. You are asked if you want a typical or custom installation. A typical installation installs all components of ObjectStore and configures the host machine to run both ObjectStore clients and server. For this procedure, it is assumed that you are doing a typical installation. If you want to do a custom installation, see Custom Installation on page 26.
6. The `setup` program displays the information you have entered, including the ObjectStore components that will be installed. Click OK to proceed with the installation.

   At this point, `setup` creates the directory structure described in ObjectStore Directory Structure and proceeds to install the ObjectStore components in this directory structure.

7. After installing the components, `setup` asks if you want to change any of the ObjectStore server parameters. Unless you know from previous experience that you will have to change any of the settings, you should not need to change them at this time.
Installing ObjectStore

8 You are asked if you want to create rawfs partitions. For more information, see What is a rawfs? on page 14

9 You are asked if you want to initialize the server log file (also known as the transaction log). You should initialize the transaction log only if you have never run an ObjectStore server process on this host before. It must be initialized before you can start up the server for the first time. For more information, see What is the Transaction Log? on page 14

10 If you answered Yes in the previous step, setup asks for the name and path of the transaction log. If you do not want to use the default, you should specify a location where the transaction log is not likely to be moved or deleted accidentally. For example, do not put it in a temp directory. Also, if you have multiple physical disk drives and need maximum performance, put the transaction log on a different disk from where you have located the ObjectStore database. Locating the transaction log on a different disk results in faster propagation of the transaction log.

11 You are asked if you want the ObjectStore server process to start up automatically at system startup. Technical Support recommends configuring the server for automatic startup. You can always shut down the server and restart it manually.

12 You are asked if you want to start the ObjectStore cache manager and (if you specified automatic startup for the server in the previous step) the ObjectStore server at this time.

13 You are asked if you want to view the README file.

At this point, the typical installation is complete.

Custom Installation

The following procedure is for users who want to select which of the ObjectStore components to install and to customize the configuration.

Note

If you are upgrading, you must first uninstall the currently installed release of ObjectStore, as described in Uninstalling ObjectStore. The following procedure assumes that you do not have ObjectStore currently installed on the target machine:

1 Run the setup program that is included in the installation package.

2 The setup program displays the License Agreement and asks you to indicate your acceptance of the Agreement by clicking Yes. The program next displays the copyright notice and asks you to indicate your acknowledgement by clicking OK.
Chapter 3: Installing and Configuring ObjectStore

3 You are asked to supply user information. If setup can retrieve this information from the Windows registry, it will display it.

4 You are asked to enter the path of the destination directory where you want ObjectStore to be installed.

5 You are asked if you want a typical or custom installation. This procedure assumes that you want to do a custom installation. If you want to do a typical installation, see Typical Installation on page 25.

6 At this point, the setup program displays all of the ObjectStore components and asks you to deselect those that you do not want installed. A check next to a component’s name means that it will be installed. By default, all the ObjectStore components are checked. Uncheck the components in the left pane that you do not want to install.

   - If you have only a C++ application environment, uncheck Java Interface and JMTL, and leave DDML checked if your application might require it.

   - If you have only a Java application environment, leave both the C++ and Java Interfaces checked and leave JMTL or DDML checked if your application might require it.

   - If you have a Java and C++ application environment, you should uncheck JMTL and DDML if your application does not need these components.

Note that, if you click on any of the component names listed in the dialog box, setup will display a brief description of the selected component.

Do not check or uncheck anything in the right pane and click Next.

7 You are asked which of the following you want to install on this machine:

   - Client run time only.
   - Server run time only.
   - Client and server run time.

Depending on your choice, some of the remaining steps will occur and others will not.

8 (Client-Only configuration.) You are asked for the name of the remote directory that contains the ObjectStore application schema database files. Unless you know the location of this directory, you should leave the path-information field blank and click Next. The setup program will display a message saying that it cannot resolve the path. Click Yes to indicate that you will resolve the path at a later time. For information about relocating
and resolving the application schema database files, see Resolving Client-Only Schema Database Issues on page 40.

9 You are asked for the name the folder where you want setup to add shortcuts. The default folder is ObjectStore.

10 The setup program displays (among other things) a list of the components it will install. Click the Back button if you want to change any of the listed information; otherwise, click the Next button to continue with the installation.

11 At this point, setup installs the ObjectStore components that you have selected.

12 (Server-Only and Server-and-Client configurations.) After installing the components, setup asks if you want to change any of the ObjectStore server parameters. Unless you know from previous experience that you will have to change any of the settings, you should not need to change them at this time.

13 (Server-Only and Server-and-Client configurations.) You are asked if you want to create rawfs partitions. For more information, see What is a rawfs? on page 14

14 (Server-Only and Server-and-Client configurations.) You are asked if you want to initialize the server log file (also known as the transaction log). You should initialize the transaction log only if you have never run an ObjectStore server process on this host before. It must be initialized before you can start up the ObjectStore server process for the first time.

15 (Server-Only and Server-and-Client configurations.) If you answered Yes in the previous step, setup asks for the name and path of the transaction log. If you do not accept the default, you should specify a location where the transaction log is not likely to be moved or deleted accidentally. For example, do not put it in a temp directory. Also, if you have multiple physical disk drives and need maximum performance, put the transaction log on a different disk from where you have located the ObjectStore database. Locating the transaction log on a different disk results in faster propagation of the transaction log.

16 (Server-Only and Server-and-Client configurations.) You are asked if you want the ObjectStore server process to start up automatically at system startup. Technical Support recommends configuring the server for automatic startup. You can always shut down the server and restart it manually.
17 If you answer Yes to this question, you are also asked if you want to start the server to start now. It is a good idea to start the server now so that you can verify its operation after it is installed.

18 *(Client-Only and Server-and-Client configurations.)* You are asked if you want the cache manager to start up automatically and, if so, whether you want it to start now. Technical Support recommends answering Yes to both questions for the same reasons given in the previous step.

19 You are asked if you want to view the *README* file.

At this point, the custom installation is complete.

### ObjectStore Environment Variables

The ObjectStore installation assigns default values to several environment variables. Normally, you do not need to change the default settings of these environment variables. The default settings are shown below.

- **INCLUDE**
  - Include path for C++ API:
    - `%OS_ROOTDIR%\include` (if C++ API is used)
    - `%OS_ROOTDIR%\..\osji\include` (if OSJI Java/C++ interoperability is used)
    - `%OS_ROOTDIR%\..\ddml\include` (if DDML is used)

- **LIB**
  - Library path for C++ API:
    - `%OS_ROOTDIR%\lib`
    - `%OS_ROOTDIR%\..\osji\lib` (if Java Interface is used)
    - `%OS_ROOTDIR%\..\ddml\lib` (if DDML is used)

- **OS_ROOTDIR**
  - Installation directory for ObjectStore (ends with `ostore`)

- **OS_TMPDIR**
  - Location of log files used by ObjectStore services
Verify Installation

**PATH**
Path to ObjectStore executables (services and utilities):

- %OS_ROOTDIR%\bin
- %OS_ROOTDIR%\..\osji\bin (if Java Interface used)
- %OS_ROOTDIR%\..\ddml\bin (if DDML is used)
- %OS_ROOTDIR%\..\jmtl\bin (if JMTL is used)

**CLASSPATH**
Path to OSJI and JMTL classes:

- %OS_ROOTDIR%\..\osji\osji.jar (if Java Interface is used)
- %OS_ROOTDIR%\..\osji\tools.jar (if Java Interface is used)
- %OS_ROOTDIR%\..\osji\osjdo.jar (if JDO is used)
- %OS_ROOTDIR%\..\osji\jdo.jar (if JDO is used)
- %OS_ROOTDIR%\..\jmtl\jmtl.jar (if JMTL is used)
- %OS_ROOTDIR%\..\jmtl\xerces.jar (if JMTL is used)

---

**Verify Installation**

After you install the ObjectStore software, you can verify that the installation process properly configured your environment. Select the processes that are appropriate for the type of installation you specified:

- Verify Installation Directory Contents on page 31
- Verify ObjectStore Services on page 32
- Verify a C++ Development Client on page 33
- Verify a C++ Run-time Client on page 33
- Verify a Java Client on page 34
- Configuration Tasks on page 34
Verify Installation Directory Contents

Your installation directory contents will depend on what you have selected to install. Some directories will always be installed. You should verify that the following directories or subset of them have been installed:

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ostore</td>
<td>Directory for the ObjectStore DBMS, and C++ environment; also contains setup.exe for performing configuration changes</td>
</tr>
<tr>
<td>ostore\bin</td>
<td>All ObjectStore utilities (.exe files) and DLLs</td>
</tr>
<tr>
<td>ostore\binsngl</td>
<td>All utilities and DLLs for an ObjectStore Single environment</td>
</tr>
<tr>
<td>ostore\etc</td>
<td>Configuration files</td>
</tr>
<tr>
<td>ostore\examples</td>
<td>ObjectStore C++ API code examples</td>
</tr>
<tr>
<td>ostore\include</td>
<td>ObjectStore C++ API include files</td>
</tr>
<tr>
<td>ostore\lib</td>
<td>All ObjectStore libraries and schema databases</td>
</tr>
<tr>
<td>ostore\unsupported</td>
<td>ObjectStore performance measurement tools which are available for customers to use, but are not a supported component of ObjectStore</td>
</tr>
<tr>
<td>doc</td>
<td>Documentation in HTML/PDF for all ObjectStore components</td>
</tr>
<tr>
<td>osji</td>
<td>This directory is for the ObjectStore interface.</td>
</tr>
<tr>
<td>osji.jar</td>
<td>ObjectStore Java interface (OSJI) classes.</td>
</tr>
<tr>
<td>tools.jar</td>
<td>OSJI development tools classes.</td>
</tr>
<tr>
<td>osjdo.jar</td>
<td>ObjectStore JDO (OSJDO) implementation classes.</td>
</tr>
<tr>
<td>jdo.jar</td>
<td>Standard JDO interface.</td>
</tr>
<tr>
<td>jdd.jar</td>
<td>ObjectStore’s Java Dynamic Data classes.</td>
</tr>
<tr>
<td>browser.jar</td>
<td>ObjectStore Java Browser.</td>
</tr>
<tr>
<td>stublib.jar</td>
<td>Stubs of OSJI classes that allow user-defined persistence-capable classes to be used in a pure transient manner. For details, see “Using Persistent Classes in a Transient Manner” in Chapter 13 (Miscellaneous Information) in the Java API User Guide.</td>
</tr>
</tbody>
</table>
Verify Installation

The debug versions of the OSJI jar files. They are compiled with the -g flag and have symbolic information that is useful to ObjectStore engineers. If you are having a problem, you might be asked to use these jar files to obtain a stack trace.

- osji_g.jar
- tools_g.jar
- jdd_g.jar

osji\lib
Directory of libraries and library schema databases.

osji\bin
Directory of executables, DLLs, and batch files

osji\include
Directory of include files needed for C++ interoperability.

osji\com\odi\demo
Demonstration hierarchy of examples of ObjectStore programs.

osji\com\odi\tutorial
Instructions and sample program for getting started.

osji\setup.bat
Script that initializes OSJI after it is installed.

osji\Setup.class
Run by the setup script for client-only installations.

osji\run_person.bat
Script to verify the installation by running the Person demo.

jmtl
Installation directory for the Java Middle Tier Library; contains the .jar files for using JMTL.

jmtl\bin
Utilities.

jmtl\examples
Directory of JMTL examples.

ddml
Installation directory for the Dynamic Data Modeling Library.

Verify ObjectStore Services

For a typical or client and server installation, you should verify that both the ObjectStore server process and the ObjectStore cache manager process are listed in the collections of services for your machine. The list of services is displayed by selecting Control Panel | Administrative Tools | Services. You should make sure that you can start and stop these services. By default the services are started automatically if you chose this option during installation.
Verify a C++ Development Client

If you have a development environment on this host with a VC++ compiler, you can build and run the `hello2` example by going to `ostore\examples\hello2` and typing:

doall.bat

Building and running this example verifies that your development environment is set up and that you can create an ObjectStore database with the C++ API.

Verify a C++ Run-time Client

If you set up a C++ run-time environment, you will not be able to build a C++ application, but simply run one on this host. A simple way to verify that a C++ run-time client is properly installed is to run one of the utilities that come with ObjectStore. In the verification procedure shown below, the ObjectStore `ossize` utility is used.

1. In a default installation, the necessary environment variables are set correctly. However, you may want to make sure that the `OS_ROOTDIR`, `LIB`, and `PATH` environment variables are set to the values shown in ObjectStore Environment Variables on page 29.

2. Run the `ossize` utility that comes with ObjectStore in the `ostore\lib` directory on an ObjectStore database. You can use any of the library schema databases found in the `ostore\lib` directory. These are the `*.adb`, `*.db`, or `*.ldb` files.

   The actual location of these database files depends on which host the ObjectStore server process is running and if the ObjectStore server process has been configured for rawfs database access. If the host for the ObjectStore server process has been configured for rawfs databases and the library schema databases are in the default rawfs for this host, then you can verify the client on this host (whether a UNIX or Windows host) as follows:

   `ossize <RemoteHostName>::/ostore/schema/6.2/metaschm.db`

   If the host with the ObjectStore server process running has a different location for the library schemas, then you will need to enter that path, rather than the default path shown above.

   If the remote host is not configured for rawfs databases, then you must find out where ObjectStore has been installed on that host.
Configuration Tasks

Here is an example of how to access the library schema on the remote host that is UNIX assuming ObjectStore has been installed in /usr/local:

```
ossize <RemoteHostName>:/usr/local/ODI/ostore/lib/metaschm.db
```

Here is an example of how to access the library schema on the remote host that is Windows assuming ObjectStore has been installed on C:\:

```
ossize \<RemoteHostName>c:\ODI\ostore\lib\metaschm.db
```

Verify a Java Client

To test the installation, enter the following command in the %OS_ROOTDIR%\..\osji installation directory:

```
run_person
```

This runs the Person demo. It displays diagnostic messages if it detects something wrong in the installation.

If you are not running osserver on the machine on which the %OS_ROOTDIR%\..\osji directory is stored, specify a pathname for a database that resides on a machine that is running osserver.

For example:

```
setup \usr1\bob
run_person \usr1\bob\person.odb
```

As with the setup command, you can specify a host-qualified pathname.

If you are running a client with a remote ObjectStore server that requires a user name and password (authentication required), you can specify them like this:

```
setup jackhammer:c:\bob
run_person jackhammer:c:\bob\person.odb bob PassWd
```

Configuration Tasks

After you have installed ObjectStore, you can use the setup installation program (%OS_ROOTDIR%\setup.exe) to do any of the following tasks. In order to perform any of these tasks, you must first shut down ObjectStore services. The installation program prompts you to do this.

Here are the configuration tasks that setup can do:

- Change the settings of ObjectStore server parameters.
Chapter 3: Installing and Configuring ObjectStore

- Initialize an existing or new transaction log.
- Enable or disable automatic startup for the ObjectStore server.

**Note**

You must have previously installed ObjectStore before you can use the `setup` program in configuration mode; see Uninstalling ObjectStore on page 39.

To use the `setup` program to configure ObjectStore, perform the following procedure:

1. Use the Windows Start menu to navigate to the ObjectStore folder, and select the `setup` program.
2. The `setup` program asks you to select one of the following task items:
   - ObjectStore Setup.
   - Reinstall.
   - Uninstall.
   Select *ObjectStore Setup* to configure ObjectStore.
3. You are asked if you want to change any server parameters. If you answer Yes, `setup` displays a dialog box that lists the server parameters and enables you to edit their settings. For detailed information about all of the server parameters, see *Managing ObjectStore*, Chapter 2 (“Server Parameters”). Any changes you make to the server parameters will not take effect until the ObjectStore server process is restarted.
4. You are asked if you want to create rawfs partitions. For more information, see What is a rawfs? on page 14.
5. You are asked if you want to initialize the server log file (also known as the *transaction log*). You should initialize the transaction log only if you have never run an ObjectStore server process on this host before. It must be initialized before you can start up the ObjectStore server process for the first time. For more information, see What is the Transaction Log? on page 14.
6. If you answered Yes in the previous step, `setup` asks for the name and path of the transaction log. If you do not accept the default, you should specify a location where the transaction log is not likely to be moved or deleted accidentally. For example, do not put it in a *temp* directory. Also, if you have multiple physical disk drives and need maximum performance, put the transaction log on a different disk from where you have located the ObjectStore database. Locating the transaction log on a different disk results in faster propagation of the transaction log.
Upgrading from ObjectStore 6.0 or 6.1

You are asked if you want the ObjectStore server to start automatically. If you answer Yes, the server will be started automatically on system startup. If you answer No, you must start the server manually.

Upgrading from ObjectStore 6.0 or 6.1

ObjectStore C++ databases created with 6.1 are compatible with release 6.2 if their schemas were generated with Visual C++ 7. Existing 6.0 and 6.1 databases generated with Visual C++ 6, are compatible with 6.2 if the applications do not use packing pragmas with persistently stored classes and if the applications do not use integral extension types such as _int16 and _int32 in any context that appears in the schema.

If your 6.0 or 6.1 databases use schemas generated with Visual C++ 6 from code that uses packing pragmas or integral extension types, see the ObjectStore Release Notes for information about how to handle these incompatibilities.

Before you upgrade your ObjectStore installation to 6.2, you should read the README.htm and the Release Notes to make sure that there are no source code changes that must be made to the application prior to the upgrade. Ensure that the desired platform configuration is supported for this release.

Follow these steps to upgrade your system from ObjectStore 6.0 or 6.1 to ObjectStore 6.2:

1. Back up all databases, application files, and the ObjectStore installation directory.
2. Shut down all ObjectStore applications or services that are running on or accessing the host targeted for installation. The ObjectStore cache manager process must also be shut down via Services Control Panel if it is running.
3. Once you have insured that no applications are accessing the ObjectStore server process, then checkpoint the ObjectStore server process to make sure that all data is propagated from the ObjectStore transaction log to the application databases. You can do this as follows:
   ossvrchkpt <hostname>
4. Shut down the ObjectStore server process via the Services Control Panel.
5. Follow the instructions in Installing ObjectStore on page 24.
6. Perform the verification as described in Verify Installation on page 30.
Lastly, you should verify that your application runs as expected after the upgrade.

Upgrading from Release 5.1

The internal database format changed between 5.1 and 6.2. This change was implemented with 6.0. This means that 5.1 ObjectStore clients and 6.2 ObjectStore clients cannot access the same database. In order for a 6.2 ObjectStore client to access a database created by a 5.1 client, the database needs to be dumped and loaded with the ObjectStore utilities `osdump` and `osload`.

This type of upgrade can be summarized with the following steps. You should consult the Migration Guide (www.ObjectStore.net/documentation/migration) for detailed instructions on upgrading this release of ObjectStore.

1. Install the 5.1 `osdump` utility that is compatible with ObjectStore 6.2.
2. Install ObjectStore 6.2 in a directory different from that of 5.1. See Installing ObjectStore on page 24.
3. If the application is written in C++, read the Migration Guide and migrate your application accordingly.
4. Completely rebuild your application (C++ migrated code or Java code) with ObjectStore 6.2 and using an ObjectStore 6.2 server process.
5. Back up your 5.1 application and database(s).
6. Dump your ObjectStore 5.1 database(s) using the 5.1 `osdump` utility from step 1 using the ObjectStore 5.1 server process.
7. Load your database(s) using the 6.2 `osload` utility.
8. Test your 6.2 application completely.
Upgrading from Release 5.1
Chapter 4
Miscellaneous Information

Uninstalling ObjectStore

You can use the following procedure to uninstall ObjectStore:

1. Shut down all ObjectStore client applications that are running.
2. Shut down all ObjectStore Services that are running.
3. Use the Windows Start menu to navigate to the ObjectStore folder, and select the setup program.
4. When the setup program starts to run, it detects that ObjectStore is installed and running, and asks you to select one of the following options:
   - ObjectStore Setup.
   - Reinstall.
   - Uninstall.
   Select Uninstall to uninstall ObjectStore.
5. The setup program lists the ObjectStore components that will be uninstalled and asks you to confirm that you want to uninstall ObjectStore.
6. If you answered Yes in the previous step, the setup program asks if you want to delete the contents of any rawfs partitions. For more information, see What is a rawfs? on page 14.
7. You are asked once more to confirm that you want to continue with the uninstall process. If you answer Yes, the setup program will stop all ObjectStore services and completely uninstall ObjectStore.
You can add ObjectStore 6.1 components to an existing ObjectStore 6.1 installation using the ObjectStore CDROM. Follow this procedure to add a component:

1. Mount the CDROM containing the ObjectStore installation package.
2. Open a command window and change your working directory to the sub-directory on the CDROM containing the component you wish to install. For example:
   ```bash
   cd jmtl
   ```
3. From the command window, run the `setup.exe` application found in the sub-directory containing the component you want to install. When running `setup.exe`, specify the pathname where you want to install the component. For example:
   ```bash
   setup.exe /z"path="c:\odi\jmtl"
   ```
   This will install the component in the current working directory to the location specified.

After you add a component, be sure to add the component information to the appropriate environment variables as described in ObjectStore Environment Variables on page 29.

If you created a client-only ObjectStore installation and did not specify the path to the remote directory containing the schema database files, you need to manually add this information to the following ObjectStore DLLs:

- `o6cmpct1.dll`
- `o6_coll1.dll`
- `o6sevol1.dll`
- `o6query1.dll`

Run the `ossetasp` utility, using the following command line to display the pathname of the current location of an application schema database file:
Chapter 4: Miscellaneous Information

ossetasp -p dll_pathname
Use the following command line to assign a new location to be used by any of the libraries listed above:

ossetasp dll_pathname new_schema_db_location
You can find detailed information about the ossetasp utility in Managing ObjectStore.

Troubleshooting Installation Problems

Sometimes on Windows NT 4.0 SP6a, you may get the following pop-up error message during installation:

![Error Message]

This error occurs when there is a problem in the Windows Install Service. In order to correct this problem you should execute InstMsiW.exe which is found with the ObjectStore installation files in the same directory as setup.exe.

After you run the InstMsiW.exe executable, you should be able to install ObjectStore.
Troubleshooting Installation Problems