Installation Configuration Guide

Version 10.3.3

SYNERGY DE

Printed: May 2016

DCN IC-01-10.3_05

Synergex
2330 Gold Meadow Way
Gold River, CA 95670 USA

www.synergex.com
phone 916.635.7300
fax 916.635.6549

Contents

Preface

1 Requirements and Considerations

3 Configuring xfServer

4 Configuring Connectivity Series

5 Redistributing Synergy/DE on Windows

Index

Preface

About this manual

This guide focuses on the installation and set-up of Synergy/DE® License Manager, xfServer, and Connectivity Series products, and provides information on troubleshooting common set-up problems. You should familiarize yourself with the contents of this manual before installing these products. This guide also includes information on operating system requirements and suggestions for configuring your system for optimum performance.

Manual conventions

Throughout this manual, we use the following conventions:

- In code syntax, text that you type is in `Courier` typeface. Variables that either represent or should be replaced with specific data are in *italic* type.
- Optional arguments are enclosed in *[*italic square brackets*]*. If an argument is omitted and the comma is *outside* the brackets, a comma must be used as a placeholder, unless the omitted argument is the last argument in a subroutine. If an argument is omitted and the comma is *inside* the brackets, the comma may also be omitted.
- Arguments that can be repeated one or more times are followed by an ellipsis…
- A vertical bar (|) in syntax means to choose between the arguments on either side of the bar.
- Data types are **boldface**. The data type in parentheses at the end of an argument description (for example, (**n**)) documents how the argument will be treated within the routine. An **a** represents alpha, a **d** represents decimal or implied-decimal, an **i** represents integer, and an **n** represents numeric (which means the type can be **d** or **i**).
- To “enter” data means to type it (or highlight it, in the case of a selection window entry) and then press ENTER. (“ENTER” refers to either the ENTER key or the RETURN key, depending on your keyboard.)

WIN

- Items or discussions that pertain only to a specific operating system or environment are called out with the name of the operating system.

Other resources

- *Synergy DBL Language Reference Manual*
- *Environment Variables & System Options*
- *Synergy Tools*
- *SQL Connection Reference Manual*
- *xfODBC User's Guide*
- *xfNetLink & xfServerPlus User's Guide*

Product support information

If you cannot find the information you need in this manual or in the resources listed above, you can reach the Synergy/DE Developer Support department at the following numbers:

800.366.3472 (in the U.S. and Canada)
916.635.7300 (in all other locations)

To learn about your Developer Support options, contact your Synergy/DE account manager at one of the above numbers.

Before you contact us, make sure you have the following information:

- The version of the Synergy/DE product(s) you are running
- The name and version of the operating system you are running
- The hardware platform you are using
- The error mnemonic and any associated error text (if you need help with a Synergy/DE error)
- The statement at which the error occurred
- The exact steps that preceded the problem
- What changed (for example, code, data, hardware) before this problem occurred
- Whether the problem happens every time and whether it is reproducible in a small test program
- Whether your program terminates with a traceback, or whether you are trapping and interpreting the error

Synergex Professional Services Group

If you would like assistance implementing new technology or would like to bring in additional experienced resources to complete a project or customize a solution, Synergex® Professional Services Group (PSG) can help. PSG provides comprehensive technical training and consulting services to help you take advantage of Synergex's current and emerging technologies. For information and pricing, contact your Synergy/DE account manager at 800.366.3472 (in the U.S. and Canada) or 916.635.7300.

Comments and suggestions

We welcome your comments and suggestions for improving this manual. Send your comments, suggestions, and queries, as well as any errors or omissions you've discovered, to doc@synergex.com.

1

Requirements and Considerations

Windows Requirements

Supported platforms

Synergy/DE 10.3.3 is supported on the following Windows platforms. Both 32-bit and 64-bit are supported on most platforms, with exceptions noted below.

- Windows 10
- Windows 8 and 8.1
- Windows 7 SP1
- Vista SP2
- Server 2012 R2 (64-bit only)
- Server 2012 (64-bit only)
- Server 2008 R2 SP1 (64-bit only)
- Server 2008 SP2

Patches

Regardless of Windows platform, you should always load all Windows critical updates.

Microsoft .NET Framework requirements

Microsoft .NET Framework version 4.0 or higher must be installed before you can install Synergy/DE version 10.3. On Windows 8 and higher and on Server 2012, version 4.*x* is installed by default, but you may need to separately install it on other Windows platforms. This is required for the Synergy .NET assembly API.

Framework 4.0 or higher is also required for the following:

- xfNetLink .NET
- Synergy DBL Integration for Visual Studio
- To develop Synergy .NET applications

Framework 4.5 or higher is required for the following:

- Synergy .NET development for Mono on Android
- Device licensing

Note that the version of the Framework installed by default on Window 8 and higher is a minimal version, intended only for running .NET applications. For .NET development, you will need the *full* version, which is installed with Visual Studio.

When running Synergy .NET applications, you'll need to install the version of the .NET Framework that your application targets.

Minimum processor requirements

The minimum requirement for Synergy/DE version 10.3.3 is a 32-bit machine with an Intel Pentium 4 or Athlon 64 processor supporting SSE2.

Environment setup

The installation program installs a default **synergy.ini** file in the DBLDIR directory. You should copy this **synergy.ini** file to another location and set the SFWINIPATH environment variable to point to it. You can then add environment settings to the file as necessary.

For more information on **synergy.ini**, see "Synergy initialization files" in the "Environment Variables" chapter of *Environment Variables & System Options*. For information on setting SFWINIPATH, see SFWINIPATH in that same chapter. For specifics on environment setup on 64-bit Windows, see the "Synergy/DE 10 on 64-bit Windows" paper in the Resource Center on the Synergex website.

Installation considerations

Before installing Synergy/DE, you should consider whether you want the software to reside locally on each machine or reside on a shared machine accessed by one or more clients. These two choices are mutually exclusive: components can either be run locally *or* accessed from a shared drive.

⚠ To run a Synergy .NET application, you must have Synergy/DE installed locally unless you are deploying a Windows Store or ClickOnce application.

A shared configuration makes maintenance easier because all Synergy/DE files are installed on the shared machine. Although you may occasionally have to upgrade the client machines, in general you will need to upgrade only the shared machine.

See the installation instructions that accompanied your distribution for detailed instructions on installing Synergy/DE to run locally or from a shared drive.

In addition, you should consider how you want to license your Synergy/DE software. Synergy/DE software is licensed with License Manager, which can be configured for local licensing or network licensing.

Running Synergy/DE from a shared machine

The shared machine can be any of the supported Windows platforms listed in “Supported platforms” on page 1-2.

The client machines can be any of the following Windows platforms:

- Vista
- Windows 7
- Windows 8 and 8.1
- Windows 10

The client cannot be a terminal server machine, which includes Windows Server 2008, 2008 R2, 2012, and 2012 R2.

The client may be a 32-bit Synergy/DE client installed on a 64-bit machine.

Licensing

If you plan to run Synergy/DE locally, you can use either local or network licensing. You’ll be prompted to choose the license type during installation. If you plan to run Synergy/DE from a shared machine, you must use network licensing.

Local licensing uses License Manager to manage licenses on a single machine. License configuration keys must be installed on each machine.

Network licensing uses License Manager server to manage licenses for multiple machines (license clients). With network licensing, you are required to install configuration keys only on the license server machine. The license clients then obtain their licensing from that machine.

See “Deciding how you want to configure License Manager” on page 2-3 for more information.

UNIX Requirements

Supported platforms

Synergy/DE 10.3.3 is supported on the following UNIX platforms.

Platform	System code	Version
HP-UX 32-bit (PA-RISC)	009	11i v3 or higher
HP-UX 64-bit (PA-RISC)	309	11i v3 or higher
HP-UX 64-bit (Itanium)	509	11i v3 or higher
IBM AIX 32-bit	004	6.1 or higher
IBM AIX 64-bit	304	6.1 or higher
Linux (x86)[a] [b]	028	CentOS 6.5 or higher Debian 7 or higher Fedora 13 or higher Red Hat RHEL 6 or higher SUSE 11.3 or higher Ubuntu 11.10 or higher
Linux (x64)[a]	428	
Oracle Solaris 32-bit (SPARC)	020	10 or higher
Oracle Solaris 64-bit (SPARC)	320	10 or higher
Oracle Solaris 64-bit (x86-64)	420	10 or higher

a. If you plan to enable SELinux, see Synergex KnowledgeBase article 100002043. xfServer and xfServerPlus are not supported on LDAP-enabled Linux systems.

b. For 32-bit Linux, the minimum processor requirements are a Pentium 4 or Athlon 64 processor supporting SSE2.

Patches

Platform	Patch	Notes
Linux (x86 and x64)	libtinfo.so.5	This library is required by the runtime.
	libssl.so.1.0.0	This library is required for OpenSSL—used by xfServer/xfServerPlus encryption, data encryption, and Synergy HTTP document transport API. See also "OpenSSL Requirements" on page 1-11.
Oracle Solaris (SPARC) v10	119963-23 118833-36	119963 is a shared library patch for the C++ runtime. 118833 is a kernel patch. Both are required by the compiler. Solaris patches can be downloaded from the My Oracle Support section of the Oracle website at support.oracle.com/CSP/ui/flash.html. Be sure to get the most recent revision of the patch. The first six digits are the patch number and the two digits following the hyphen are the revision. More recent revisions than those listed here may be available.
Oracle Solaris (x86) v10	119964-23 118855-36	119964 is a shared library patch for the C++ runtime. 118855 is a kernel patch. Both are required by the compiler. See the information above (in the Notes column for SPARC v10) regarding downloading Solaris patches.

Environment setup

The Synergy environment must be set up for each user who wants to run Synergy/DE products. The necessary environment variables can be set at either the system level or the user level.

⚠ Source the **setsde** script (located in the synergyde directory) to enable the Synergy/DE environment for the products you have installed and licensed. For example,

```
. ./setsde
```

You may want to add this script to your start-up file. **Setsde** can be used with any UNIX shell except **csh**. Synergex uses **ksh**; if you do not use **ksh**, your UNIX shell may require different syntax than that included in the **setsde** script. See your system administrator for assistance.

OpenVMS Requirements

Supported platforms

Synergy/DE 10.3.3 is supported on the HP OpenVMS Alpha platform and on HP OpenVMS I64 Integrity servers.

Platform	System code	Version	Patches
Alpha	200	7.3-2 or higher[a]	VMS*version*_ACRTL-V*####* VMS*version*_RMS-V*####* VMS*version*_UPDATE-V*####* where *version* is the OpenVMS version and *####* indicates the version of the patch. You should install the most recent version of the patch available for your version of OpenVMS. For example, for OpenVMS 7.3, the patches might be VMS73_ACRTL-V0600, VMS73_RMS-V0400, and VMS73_UPDATE-V0300.
I64	202	8.3 or higher[a]	Download and install the most recent version of the C++ runtime patch VMS*version*_ICXXL-V*####*, where *version* is the OpenVMS version and *####* indicates the version of the patch, for example, VMS83I_ICXXL-V0500. You will not be able to install Synergy/DE without first installing this patch!

[a] Version 8.4 is required for SSL1 1.0-2C compatibility. See "OpenSSL Requirements" on page 1-11 for more information. Because Connectivity Series supports only SSL1 1.0-2C, OpenVMS version 8.4 or higher is required for encryption support with Connectivity Series.

Patches

The OpenVMS patches listed in the table above are available for download from the Hewlett-Packard website. To find the most recent version of a patch, do a partial search on the patch name (e.g., search on VMS73_ACRTL).

TCP/IP (Alpha)

TCP/IP version 5.0a ECO2 or higher is required on OpenVMS Alpha. We recommend using the most recent version of TCP/IP available for your version of OpenVMS.

To check the TCP/IP version on your machine, execute this command:

```
$ tcpip show version
```

Sample output:

```
HP TCP/IP Services for OpenVMS Alpha Version V5.4 - ECO 5
on a AlphaServer DS10 617 MHz running OpenVMS V7.3-2
```

If you’re having problems resolving an IP address in a subnet, install the most recent ECO (patch) for your version of TCP/IP. These patches are available on the Hewlett-Packard website. For example, for TCP/IP version 5.4, the most recent is ECO 7, which is named DEC-AXPVMS-TCPIP-V0504-15ECO7-1.

Environment setup

In a default installation, you must ensure that the system start-up command procedure invokes **SYS$MANAGER:SYNERGY_STARTUP.COM** to set up the environment each time the system starts up.

In an alternate installation, you must run the command file **DBLDIR:ACTIVATE_SDE.COM** manually every time you log on to set up the environment.

Licensing

The OpenVMS License Management Facility is used to license Synergy/DE on OpenVMS instead of Synergy/DE License Manager. The runtime supports cluster-wide licensing. Synergy/DE 10 license PAKs must be installed and loaded before installing Synergy/DE. To obtain Synergy/DE PAKs, complete and return the Synergy/DE Product License Agreement. For more information about PAKs, see the *OpenVMS License Management Utility Manual* or your system administrator.

Memory

The Synergy/DE compiler is designed to take advantage of memory to increase performance. If your development machine has less than 64 MB of memory, we recommend that instead of **dbl**, you use the compiler distributed as **dbl8**. (See "The dbl8 compiler" in the "Building and Running Synergy Applications" chapter of *Synergy Tools* for more information on **dbl8**.)

For maximum performance and to avoid "exceeded quota" errors, you must ensure that the account quotas for developers using the new compiler are set correctly. We recommend the following account quotas in the Authorize utility:

Authorize parameter	Pagelets
BYTLM[a]	300,000
ENQLM	2,000 or higher
FILLM[b]	500 or higher
PGFLQUO	200,000 or higher
WSEXTENT	64,000 to 120,000
WSQUOTA	32,000 up to the value of WSEXTENT

[a] To use the Synergy Prototype utility (**dblproto**) on large files (especially if they contain many import statements and .INCLUDE directives), you may need to increase the BYTLM value.

[b] When setting FILLM, verify that the CHANNELCNT SYSGEN parameter is higher than the FILLM value.

Network Requirements (TCP/IP) for Client/Server Products

Synergy/DE client/server software (SQL OpenNet, xfServer, xfServerPlus) all use TCP/IP for data transport. On Windows, License Manager server also uses TCP/IP to communicate.

When installing Synergy/DE client/server products, you need to have a thorough understanding of the installation, configuration, and troubleshooting of TCP/IP on the targeted client and server machines. Synergy/DE Developer Support cannot provide assistance with TCP/IP configuration issues. However, we provide a utility, **synxfpng**, for troubleshooting connections to xfServer, SQL OpenNet, and License Manager server.

Before any client/server communication can occur, the client has to be able to locate and then connect to the host machine. You can use either an IP address or a host name to identify the host machine (except for a configuration that includes backup license server; see “Configuring a backup license server” on page 2-11). If there is a firewall between the client and the server, it will have to be configured to allow communication on the specified port.)

If you use a host name to identify the server, either the **etc/hosts** file is used or the client machine performs a request to a DNS (Domain Name Server—a machine that all clients use to convert names to IP addresses). Machines are usually configured to use a primary DNS and, if the lookup fails there, try a secondary DNS. The fail over time (the length of time the machine will look for the DNS before reporting a failure) can be several minutes.

A typical problem at installation time is that connections are very slow. This is usually due to a DNS lookup problem. You can use **synxfpng** to test this. See “The synxfpng Utility” on page 3-58 for details.

For OpenVMS TCP/IP requirements, see “TCP/IP (Alpha)” on page 1-8.

OpenSSL Requirements

This section lists the required versions of OpenSSL for the various operating systems.

For additional information about OpenSSL, see Synergex KnowledgeBase article 100001979.

The following Synergy/DE features require OpenSSL. Before installing OpenSSL, consult the documentation pertaining to the feature you plan to use for details on where the OpenSSL libraries should be installed and whether you need to set any environment variables or paths.

- Synergy HTTP document transport API to send and receive documents via HTTPS. See the "Synergy HTTP Document Transport API" chapter in the *Synergy DBL Language Reference Manual*.
- Data encryption. See "Data encryption" in the "Welcome to Synergy DBL" chapter of the *Synergy DBL Language Reference Manual.*
- Client/server encryption with xfServer or xfServerPlus. For xfServer, see "Using Client/Server Encryption" on page 3-7. For xfServerPlus, see "Using Encryption" in the "Configuring and Running xfServerPlus" chapter of the *xfNetLink & xfServerPlus User's Guide*.
- Data packet encryption with SQL OpenNet. See "Using Data Packet Encryption for SQL OpenNet" on page 4-3.

Windows

Synergy/DE 10.3.3 supports OpenSSL version 1.0.1*x*. Connectivity Series supports version 1.*x*.

UNIX

For Synergy/DE 10.3.3, you should use the version of OpenSSL that comes from the vendor and is specific to your UNIX platform and version. The OpenSSL shared libraries required by Synergy/DE are **libcrypto.so.#.#.#** and **libssl.so.#.#.#**, where #.#.# is the version number, e.g., 1.0.1.

- For HP-UX (Itanium), use version 1.0.*x*.
- For HP-UX (PA_RISC), use version 0.9.8w.
- For IBM AIX, use the version installed with the operating system. It should be at least 1.0.*x*.
- For Linux, use the version installed with the operating system. It should be at least 1.0.*x*.
- For Solaris 11, install the latest version 1.0.*x* from Oracle. For Solaris 10, see Synergex KB article 100001979 for instructions on updating to 1.0.*x*.

Connectivity Series supports version 1.*x* of OpenSSL.

Synergy/DE features that use OpenSSL will display an error if the correct version cannot be found. If you are in doubt as to whether you have the correct version for the features you want to use, run one of the following commands:

- For data encryption, and client/server encryption with xfServer or xfServerPlus:

  ```
  ldd $DBLDIR/bin/synssllib.so
  ```

- For HTTPS encryption:

  ```
  ldd $DBLDIR/bin/httpslib.so
  ```

OpenVMS

Synergy/DE 10.3.3 supports both HPE SSL1 version 1.0-2C (which is based on OpenSSL 1.0.2 C) and HPE SSL 1.4 (which is based on OpenSSL 0.9.8*x*). SSL1 version 1.0-2C requires OpenVMS 8.4 or higher (on both Alpha and I64).

> Connectivity Series supports *only* HPE SSL1 version 1.0-2C, which means OpenVMS 8.4 or higher (on both Alpha and I64) is required for encryption support in Connectivity Series. The runtime support file **VTXSSL.EXE**, built with SSL1 1.0-2C, is installed.

On OpenVMS 8.4, the runtime support files **SYNSSLLIB.EXE** and **HTTPSLIB.EXE** are built with SSL1 1.0-2C to take advantage of the latest security fixes. We *strongly* recommend that you use SSL1 1.0-2C for security reasons, but if you require SSL 1.4 functionality, you can use the runtime support files built with SSL 1.4, which are distributed in SYNERGYDE$ROOT:[DBL.BIN]. To use the 1.4 files, copy **SYNSSLLIB_V14.EXE** to **SYNSSLLIB.EXE** and copy **HTTPSLIB_V14.EXE** to **HTTPSLIB.EXE**, and then replace them in the install tables by executing SYS$MANAGER:SYNERGY_STARTUP.COM or the appropriate INSTALL REPLACE commands therein.

On OpenVMS versions prior to 8.4, the runtime support files **SYNSSLLIB.EXE** and **HTTPSLIB.EXE** are built with SSL 1.4. The files for SSL1 1.0-2C (named **SYNSSLLIB_V101.EXE** and **HTTPSLIB_V101.EXE**) are also installed in SYNERGYDE$ROOT:[DBL.BIN]. Should you upgrade your operating system to 8.4, you can use SSL1 1.0-2C, by renaming these files to **SYNSSLLIB.EXE** and **HTTPSLIB.EXE**, respectively, and then replacing them in the install tables are described above.

Configuring for Performance and Resiliency

A Synergy application's performance and resiliency can often be significantly improved by careful system design and configuration. This section doesn't provide details for any particular system, but it does provide general guidelines for improving performance and achieving high availability (resiliency).

Issues with third-party products and their interactions are beyond the scope of this documentation and Synergy/DE Developer Support. A common example of this is operating system virtualization, where Synergy/DE is supported on the operating system, and the virtualization software acts as a hardware layer. We recommend that you maintain support contracts with third-party product vendors (e.g., operating system and virtualization software vendors) for assistance with these issues.

Systems for Synergy applications are configured in a number of ways: stand-alone (monolithic) systems, client/server configurations with data and/or processing on a server, and virtualized systems. A correctly configured stand-alone system with sufficient resources provides the best performance for Synergy applications. Such systems are very fast, and adding a hot standby makes them resilient. We recommend that Synergy applications use a stand-alone system when possible. If you find that your applications are pushing the system's limits (too many users, intensive end-of-year processing, etc.), improve the stand-alone system, if possible, before considering moving data or processing out to the network.

- For information that applies to all systems, see "General performance recommendations" below, "Configuring SAN for optimal data access" on page 1-18, and "Food for thought: Results from our testing" on page 1-19.
- For stand-alone systems, see also "Stand-alone system configuration" on page 1-16.
- For client/server systems, see also "Design and configuration for client/server systems" on page 1-16 and "Optimizing network data access with xfServer" on page 1-19.
- For virtualized systems, see also "Configuring virtual machines" on page 1-15.

General performance recommendations

For a Synergy application, optimization involves carefully configuring the hardware and software that support the application, and carefully designing and coding the Synergy application itself.

Keep the following in mind when configuring your system for Synergy applications:

- Make sure SAN drives are configured optimally (see "Configuring SAN for optimal data access" on page 1-18).
- Keep software current. Attempting to use old versions of Synergy/DE with current third-party tools (or vice versa) will result in poor performance if they work at all.

- Most Synergy programs are single-threaded, so they each use only one system core as illustrated in figure 1-1 on page 1-17. (All traditional Synergy programs are single-threaded; Synergy .NET programs can be single-threaded or multi-threaded.) A multi-user application, batch job, or cron job can use any core of a multi-core processor.
- Modern processors adjust the CPU speed automatically to save power. Make sure the CPU is clocking up for your Synergy application by setting the BIOS to allow the operating system to control power management. Some BIOS power management configurations favor power saving over performance, so if the core for a Synergy application is the only heavily utilized core at any given point, the BIOS may lower the CPU's clock speed, severely hindering performance.
- Always configure antivirus software to exclude scanning of **.is1**, **.ism**, and **.ddf** files, as well as any other data files you have with custom extensions.

A Synergy application's design and the way it is coded can make a big difference in the way it performs. Make sure you use established best practices for programming and note the following recommendations:

- Use OPEN and CLOSE as little as possible. For example, do not open or close channels in a loop, and wait until a channel is no longer needed in the current context before closing it. Data in the system cache is written to disk when a channel is released for a CLOSE, so unnecessary opening and closing of channels wastes disk subsystem bandwidth and can severely hinder performance for large workloads.
- With Synergy .NET, make sure that objects created in a loop are destroyed at the end of the loop. This includes Synergy alpha, integer, and decimal data types, which are implemented as objects. For example, if a loop uses arithmetic on a decimal type, a decimal object will be created for the results with every iteration of the loop, and .NET garbage collection may then consume a large percentage of CPU resources disposing of the leftover objects.
- For sequential READS, use LOCK:Q_NO_TLOCK for files opened in update mode when possible. This reduces disk overhead and unnecessary locking and allows caching of index blocks. But note that cached index blocks for the READS will not include records subsequently stored by other users.
- Use the OPTIONS:"/sequential" qualifier for OPEN to optimize sequential record access when using primary key access on a sequential file or an ISAM file ordered with **isutl**. This improves the effectiveness of operating system caching for sequential reads.
- Use the Q_EQ option for the MATCH qualifier to prevent record locking when a record is not found and to prevent the next record from being read and acquiring a lock.
- Use I/O error lists for common errors such as end of file, key not same, and record locked. I/O error lists are much faster than TRY/CATCH, especially over a network with xfServer, and they are built into READS, WRITES, and other Synergy DBL statements that have an error list argument.
- When using IOHooks, avoid performing network operations in your hook routines, such as posting a transaction to a SQL database.

- Use profiling tools to evaluate your applications’ performance and to uncover coding and design issues that could affect performance, particularly for processor-intensive applications such as programs for end-of-year processing. For Synergy .NET applications, use the Visual Studio CPU Usage tool (with managed and unmanaged code) and the Visual Studio Memory Usage tool. For traditional Synergy applications, use the Synergy DBL Profiler.

Optimally configuring and using data files can also affect performance. Note the following:

- All ISAM files should be REV6, which is aligned to 4K for large sector drives (i.e., index blocks are read and written on 4K boundaries). This means that indexes can be shallower, which reduces the overhead required to read records randomly. Unless you have SSD drives, limit indexes to no more than three levels deep. A smaller index means less disk movement for non-SSD disks. You may be able to further improve ISAM performance (especially for large files) by increasing page size from 4K to 16K.
- Use data compression and 100% packing density for ISAM files that change very little (e.g., archive files). For other ISAM files, we recommend you run **isutl -vc**, which gives you an idea of the expected compression savings for a file.
- Use **isutl** to maintain ISAM files. ISAM file indexes and data should be periodically optimized:

  Use **isutl –v** to verify files.

  Use **isutl –vi** for the information advisor, which provides details about a file’s condition and offers advice (including performance enhancements) based on file organization and content.

  Use **isutl –r** to re-index the file or **isutl –ro** to re-index the file and reorder the data.
- If you use snapshots, use the **synbackup** utility to prevent file corruption, and for performance reasons, consider limiting snapshots. (For performance troubleshooting, try turning off snapshots to see if that makes a difference.)

Configuring virtual machines

Virtual machines are inherently slower than equivalent physical systems because the virtualization software uses system resources and because cores on a virtual machine do not necessarily map directly to cores on the physical machine due to oversubscription. Consequently, on a virtualized system your Synergy application will not get as much CPU power as it would on a physical system.

When virtualizing systems, CPU usage has only a small overhead. I/O operations—including Toolkit, low-level windows, and most especially WPF-type graphics—slow down significantly compared to a physical system. If the latest virtual I/O hardware instructions are either not present or not enabled, the system may get even slower as more virtual CPU cores are added.

The following Xeon processors (or newer; these were introduced in 2010) will have VT-x, VT-d, and EPT required for acceptable I/O performance: Beckton (multiprocessor), Clarkdale (uniprocessor), Gainestown (dual-processor). In addition, using multiple dedicated server-class graphics cards associated with a virtual machine may help .NET WPF application performance.

On a multi-CPU socket machine, a virtual machine should be paired with a specific CPU socket to avoid negatively impacting performance.

Note the following:

- If you use non-SSD drives for a virtualized system, do not use shared virtual machine partitions on a physical drive.
- When using a virtual machine with xfServer, remember that the physical machine may share its network ports with other virtual machines because there may not be a network port dedicated to the virtual machine instance.

Stand-alone system configuration

A stand-alone system is defined as a multi-processor UNIX system, OpenVMS system, or 64-bit Windows server with the Remote Desktop Services (Terminal Services) role enabled. A stand-alone system can have from 2 to 6,000 users. Larger system are typically attached via fibre channel to a SAN array with SSD drives or 15,000 RPM SAS drives. Backups are generally done from the SAN array, rather than the operating system, since thousands of users may be accessing the system. There may be a hot standby system powered-up and attached to the same SAN array. If the main server crashes, the hot standby can be booted up, so the Synergy application can be back up and running within minutes. See figure 1-1 on page 1-17.

If you have a hot standby system, use system option #36 or selectively use the FLUSH statement to prevent data corruption from occurring when the hot standby is brought into service. This flushes (writes to disk) the operating system cache, which is lost when a hot standby system starts up. Flushing protects data integrity, but it can hurt write performance because operating system caching is effectively bypassed. If you use option #36 system-wide, consider disabling it with %OPTION for programs that write or update large amounts of data, such as day-end processing.

Design and configuration for client/server systems

On a client/server system Synergy data is located on a remote server and, optimally, some data processing takes place on that server as well. When you move a Synergy application from a stand-alone configuration to a client/server configuration, it is likely to need significant architectural changes to approach the performance it had when stand-alone. For example, data updates and random read statements can be 100 times slower in a client/server environment than in a stand-alone environment, especially with a WAN (see “Food for thought: Results from our testing” on page 1-19).

We recommend the following:

- Use 64-bit architecture, including 64-bit terminal server clients, when possible. Use the latest version of Synergy/DE for applications on clients, and upgrade to the latest versions of server software (exchange servers, domain controllers, SQL Server, etc.).
- Ideally, clients and servers should be connected with at least Gigabit Ethernet (GbE).

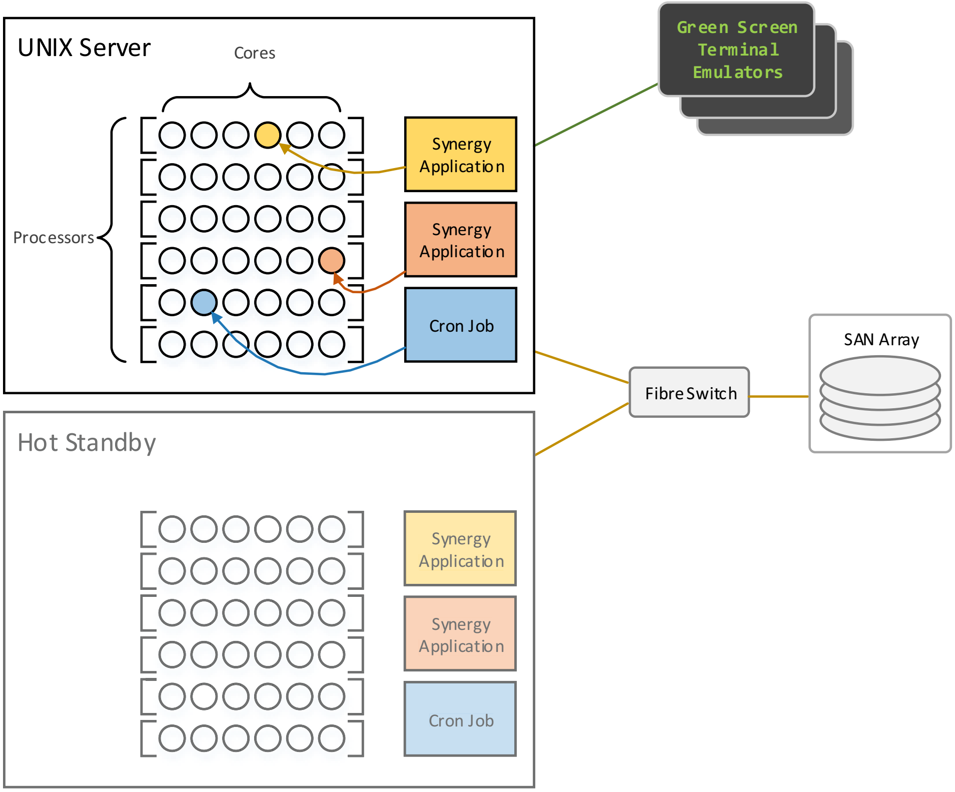


Figure 1-1. A typical stand-alone system for Synergy applications.

- Do not use mapped drives on Windows (i.e., Microsoft SMB network shares). They can cause performance problems and file corruption issues, particularly with multi-user access. Use xfServer instead. Generally speaking, xfServer significantly outperforms mapped drives in multi-user situations when it is properly configured. (See “Optimizing network data access with xfServer” on page 1-19.) Synergex can provide only very limited support for Synergy database access through mapped drives.
- When files are on a separate server, it’s best to put file processing logic on that server as well by using xfServer with the Select class or by using xfServerPlus method calls. Otherwise, the Synergy application will use up network bandwidth by passing unnecessary data over the network rather than passing only user-selected records. Programs that transfer large amounts of data across a network (e.g., month-end or day-end processing or reports) can quickly overload a network. See “Optimizing network data access with xfServer” on page 1-19.

- For network printing, connect the user’s machine directly via the network to the printer, rather than employing a print server.
- On OpenVMS, disk I/O is not cached, and xfServer and SQL OpenNet server do not scale well. Be careful when configuring OpenVMS servers for dozens of xfServer client processes or xfODBC users.

Configuring Windows terminal servers

Terminal servers are machines with the Remote Desktop Services (Terminal Services) role enabled. Do not activate this role unless it is absolutely necessary. File servers, for example, should not have this role (use a separate server for data). Note the following:

- Terminal servers should be 64-bit machines.
- Turn off Fair Share options (CPU scheduling, network, and disk options). These options are set by default and affect performance even when there is only one logged-on user or scheduled task. See “Performance Problem when Upgrading to Server 2012,” available on the Synergy/DE CTO blog site.
- For xfServer, consider using fewer terminal servers with more users on each. You may even be able to eliminate the need for xfServer by consolidating to a single 64-bit terminal server.

Configuring SAN for optimal data access

For best performance with Synergy ISAM data, we recommend using high-quality 15,000 RPM SAS disks or SSDs configured in RAID 1 mirror sets. (Other RAID configurations are not recommended.) SANs should use fibre channel or drives should be directly attached. Note the following:

- For SSDs, make sure the SCSI unmap function is supported and that the controller on the SAN port of the SAN does the appropriate trim-equivalent pass-through.
- NAS connections are not supported, and iSCSI connections are very slow.
- When tuning SAN arrays, we do not recommend using RAID 0 (striping), because Synergy uses 4K buffers for index operations with ISAM, 32K buffers for sequential operations, and 64K buffers when performing a SORT. If you do use RAID 0, consider a stripe size of 4K.
- On Windows, it is best to move data files and work files (TEMP:) to a non-system disk due to the overhead required for Windows filter manager. For example, consider using a separate virtual hard disk (VHD) with a separate logical unit number (LUN) for these files.
- Use SSD drives on a SAN if you use shared disk partitions to clients on a physical LUN.

Optimizing network data access with xfServer

xfServer provides reliable, high-performance access to remote Synergy data. It manages connections to files and file locks, preventing data corruption and loss, and it can shift the load for data access from the network to the server, easing network bottlenecks.

There are two environment variables you can use with xfServer to improve performance, SCSPREFETCH and SCSCOMPR. Setting SCSPREFETCH turns on both prefetching, in which sequential records are fetched from the server and stored in a buffer until needed, and buffering, in which records waiting to be written to the server are buffered and then written all at once. Prefetching works only on READS. Buffering works on WRITES and PUTS; an equivalent option for sequential STOREs is the OPTIONS:“/bufstore” qualifier on the OPEN statement. For details see “Prefetching and Buffering Records” on page 3-6.

Setting SCSCOMPR turns on data compression, which compresses blanks, nulls, zeros, and repeating characters in records sent between xfServer and its clients. Compression can significantly improve performance on low speed or busy networks, especially WANs.

The Select classes not only make it easier to write code that accesses Synergy data (by creating SQL-like statements), they can also improve performance because reading the ISAM files and selecting the desired records takes place on the server. Then, only the necessary records are sent over the network to the client. If you want to transmit only the necessary fields within records, you can use the Select.SparseRecord method or the Select.Sparse class. For sparse records to be truly effective in reducing network traffic, SCSCOMPR should be set as well. A corresponding method, SparseUpdate, can be used when writing data.

Use the DBG_SELECT environment variable to determine how well your Select queries are optimized. See the “System-Supplied Classes” chapter of the *Synergy DBL Language Reference Manual* for more information about the Select classes.

On a wireless network, use xfServer connection recovery (SCSKEEPCONNECT) to improve resiliency. This feature enables an xfServer client application to seamlessly reconnect to the server and recover its session context after an unexpected loss of connectivity. See “Using connection recovery” on page 3-19.

For more information about xfServer in general, see “What Is xfServer” in the “Accessing Data Remotely with xfServer” chapter of *Getting Started with Synergy/DE*.

Food for thought: Results from our testing

To illustrate how different configurations affect performance, we did some testing with similar single-user Windows and UNIX systems and an ISAM file with one million 200-byte records, one key, and data compression at 50%. (Note that although the systems were similar, they used different hardware and had different capabilities. The figures cited below are meant to show the differences between operations, not operating systems.) Here's what we found for local (non-network) access:

Operation	Records per second	
	Windows	Linux
STORE to an ISAM file	88,000	68,000
READS from an ISAM file opened in input mode	588,000	1,485,000
READS from an ISAM file opened in update mode	175,000	410,000
READS from an ISAM file opened in update mode with /sequential	330,000	657,000

The following table shows the results for xfServer and network access with Gigabit Ethernet. The first four rows show *local* xfServer access; you can compare them with *network* xfServer access (in the next five rows) to see how a physical network affects performance. Results for mapped drive access are supplied for comparison purposes only; accessing Synergy data via mapped drives is not recommended.

Operation	Records per second
Local xfServer STORE	37,000
Local xfServer buffered STORE	69,000
Local xfServer READS without prefetch	46,000
Local xfServer READS with prefetch	434,000
xfServer STORE from Linux client to Windows server	5,000
xfServer buffered STORE from Linux client to Windows server	37,000
xfServer READS from Linux client to Windows server without prefetch	6,000
xfServer READS from Linux client to Windows server with prefetch	105,000
xfServer READS from Linux client to Windows server with prefetch and compression	141,000
READS from a mapped drive with a single user	430,000
READS from a mapped drive when a file is open for update by another user	435

2

Configuring License Manager

Setting Up License Manager on Windows

License Manager is a set of utilities that controls the use of Synergy/DE products. On Windows, License Manager is a server, **synd**, which runs as the service **synlm**. Starting with Synergy/DE version 10, License Manager uses sockets (TCP/IP) to communicate. Previously, it used pipes. A version 10 license server can accept connections from version 10 license clients using sockets as well as from older clients that still use pipes. By default, License Manager runs on port 2380. See "Network Requirements (TCP/IP) for Client/Server Products" on page 1-10 for more information about TCP/IP.

License Manager can also be used to secure the applications that you create with Synergy/DE. When a product is secured with License Manager, it will not run unless License Manager is installed and has been configured to recognize that product. To build License Manager protection into your own applications, you'll need Synergy/DE Licensing Toolkit. For more information about this product, see the *Licensing Toolkit User's Guide* or contact your Synergy/DE account manager.

To use License Manager—and your Synergy/DE products—you'll need to complete these steps:

1. Decide whether you want to configure License Manager for local licensing or network licensing. See "Deciding how you want to configure License Manager" on page 2-3.
2. Install the Synergy/DE products; License Manager is included in the core components. See "Installing License Manager" on page 2-4.
3. Obtain license configuration keys from Synergex. See "Obtaining configuration keys" on page 2-5.
4. Install configuration keys to allow your Synergy/DE products to continue running after the 14-day evaluation period. See "Installing configuration keys" on page 2-7.

When you *first* install Synergy/DE on a machine that has never before had it installed, you will have unlimited evaluation use of all products for 14 days. After 14 days, you can run only the products that you have configured. Once you configure a particular product, your use of that product is limited to the number of licenses you purchased. After the 14-day evaluation period expires, if you want to evaluate a product, you must obtain a special configuration key from Synergex.

Each time you access the runtime during the evaluation period, the Synergex copyright message will display, along with the licensee name and the number of days remaining in the evaluation period. After you configure the runtime, this message no longer displays.

Deciding how you want to configure License Manager

License Manager can be configured for local licensing or network licensing.

Local licensing

Local licensing uses License Manager server (**synd**) to manage licenses on a single machine. You can configure local licensing on any Windows system. Each system with local licensing requires its own set of configuration keys, and you must configure a single-user license for each Synergy/DE product that will be run on that system. (You can easily configure multiple products in one step with a Synergy key file.) A single-user license grants authorization to any number of concurrent Synergy/DE application windows.

Network licensing

Network licensing uses License Manager server to manage licenses for one or more license clients. You can configure network server licensing on any Windows machine to manage license clients in a local area network (LAN).

There are two scenarios in which you *must* use network licensing:

- When 32-bit Synergy/DE is installed on a 64-bit machine. The license server can be the local machine if 64-bit Synergy/DE is also installed, or it can be any other (32-bit or 64-bit) machine.
- When Synergy/DE is installed to run from a shared drive. The machine on which Synergy/DE Client is installed must be configured as a license client. The license server can be any machine; it doesn't have to be the shared machine. See "Installation considerations" on page 1-3.

When using network licensing, configuration keys are installed only on the license server machine, not on each license client. On the license clients all you need to do is specify the name of the license server during installation. Synergy/DE applications running on the license client take their licenses from the license server; therefore, the license server must have the necessary licensing in place before you can run applications on the client.

Users accessing an application via remote desktop software or a terminal server session are licensed using the license settings on the terminal server machine. There is no licensing setup required on the remote client device.

You can configure a backup license server, which will take over the task of serving licenses should the primary server fail. See "Configuring a backup license server" on page 2-11.

For Synergy .NET applications (including those using ClickOnce deployment) that are deployed to systems where Synergy licensing is not configured, you must specify a license server name in the **App.Config** file. For details, see the "Runtime Settings Page" topic in Synergy/DE WebDocs.

Understanding concurrent users and license usage

A configuration key specifies the maximum number of concurrent users for each Synergy/DE product that will be run by the license server and its license clients. Synergy/DE products are licensed by seat, with the exception of SQL Connection, which is licensed by connection. (A seat may be defined as a desktop [that is, a person logged in to the system], a service, or a scheduled task.) This means that each license client can run multiple occurrences of a Synergy/DE product, but is counted as only one license by the license server. If the total number of concurrent users is exceeded, new Synergy/DE applications will not be allowed to run until the number of users has dropped below the maximum.

Installing License Manager

1. Follow the instructions included with your distribution to install the Synergy/DE products. License Manager is installed with core components.

2. When prompted for the license configuration, choose either "License Server or Stand-alone" (if the machine will be a license server or have local licensing) or "License Client" (if the machine will take its licenses from a license server).

3. If you choose "License Server or Stand-alone", in the Unique Licensee Name field enter the name that you provided to Synergex when you ordered Synergy/DE products for this machine. (This name will display in the copyright notice when you run licensed Synergy/DE products.) By default, the license server runs on port 2380. If you need to change it, you must do so from the command line. (See "Changing a license server's port" on page 2-9.)

TIP We recommend that your company establish a licensee naming standard. If your company installs Synergy/DE on multiple systems at one site, we recommend that you use the name of the company followed by the user or domain name (e.g., "ABC Computers, Mary Smith" or "ABC Computers, Tiger"). If your company installs Synergy/DE at multiple sites, we recommend that you specify the company and city or location (e.g., "ABC Computers, Gold River" or "ABC Computers, store 215").

4. If you choose "License Client", in the Server Name field specify the license server machine that this client will use. If your configuration includes a backup license server, you must specify the license server by DNS name; otherwise, you can specify the license server using either an IP

address or the DNS name. Depending on how your DNS is configured, you may be able to specify a shortened form of the name (e.g., Tiger) or you may have to specify the full DNS name (e.g., Tiger.ABCComputers.loc).

If the license server is not running on the default port (2380), specify the license server port by appending a colon followed by the port number to the end of the server name or IP address (e.g., Tiger:5555).

> If you are installing 32-bit Synergy/DE on a 64-bit machine, and 64-bit Synergy/DE is already installed with server licensing, the server name for the client will default to that of the 64-bit machine. You can change the 32-bit installation to be a license client to another machine if desired.

Obtaining configuration keys

You can use the following methods to obtain configuration keys:

- **Online.** If you are a Synergy/DE Developer Support customer and your Resource Center logon includes Online Keys access, you can obtain configuration keys using the Online Key Generation program on the Synergex website. For more information about Synergy/DE Developer Support and Online Keys access, contact your Synergy/DE customer service representative.
- **Manually.** You can print your registration information and then mail or fax it to Synergex. You can receive your keys by e-mail or fax. See “Obtaining keys by mail or fax” on page 2-6.

Obtaining keys online

To complete this procedure, you will need your Resource Center user name and password.

1. Gather the registration information for the machines in one of the following ways:
   - Display the registration information on screen from the Synergy Configuration Program: Go to the Licensing tab and click the View Installed Keys button. You can select and copy the text that displays. You can also view the registration information on the Licensing tab, but it cannot be selected and copied.
   - Print the registration information from the Synergy Configuration Program: Go to the Licensing tab and click the Print Registration Information button. See “Obtaining keys by mail or fax” on page 2-6 for instructions on completing the Registration Information dialog box. (Note that this option prints the *entire* license agreement, which includes the registration information. However, you can save the information to a file and then just print the relevant portions.)
   - Display the registration information at a command prompt: Run **lmu** with the **-b** option.
2. From the Licensing tab in the Synergy Configuration Program, click the Obtain Keys Online button to launch your web browser and go to the Resource Center. (Or, start your web browser and go to the Synergex website, then log into the Resource Center.)

3. On the main Resource Center page, click on the link for “Online Ordering, Online Keys”, and then on the link for “Generate Configuration Keys”.
4. Accept the license agreement and then follow the instructions on screen to obtain configuration keys for your Synergy products.
5. When prompted, download the Synergy key file (**.skf** file) or select to have it sent to you as an e-mail attachment. Save the key file to your hard drive; you’ll use it to install the keys (see “Installing configuration keys” on page 2-7).

Obtaining keys by mail or fax

This procedure prints a license agreement, with your registration information filled in, which you can mail or fax to Synergex.

1. Start the Synergy Configuration Program (from the Windows Control Panel, select Synergy Control Panel > Synergy Configuration Program).
2. On the Licensing tab, click the Print Registration Information button.
3. Complete the fields in the Registration Information dialog box:

   Company name. Enter the name of your company.

   Contact name. Enter the name of the person at your company to contact about licensing issues.

   Phone number. Enter the phone number of the contact person.

   Send keys to me by. Choose the method by which you want Synergex to send your configuration keys to you.

   E-mail address. If you chose to receive keys by e-mail, enter the e-mail address that you want them sent to. You will receive a Synergy key file (**.skf** file) as an e-mail attachment.

   Fax number. If you chose to receive keys by fax, enter the fax number that you want them sent to.

   Installation type. Select New if this is the first time Synergy/DE will be installed for this license. Select Upgrade if you are upgrading the license from a previous version of Synergy/DE. Select Re-install if you are re-installing the same version of Synergy/DE for this license.

   Synergy/DE version. Select the version of Synergy/DE for which you are requesting keys. If you select Other, explain in the Comments field.

   Comments. Enter any comments you have regarding this request. If you selected Other for either “Send keys to me by” or “Synergy/DE version”, explain here.
4. Click the Print button to print the license agreement. Optionally, you can click the Save to File button to save it as a text file.
5. If you printed the license agreement, sign and date it, and then mail or fax it to Synergex. When you receive your configuration keys, install them; see “Installing configuration keys” on page 2-7.

Installing configuration keys

Before installing configuration keys, ensure that the system date and time are correct, and that no Synergy applications are running.

If you have a Synergy key file

1. Start the Synergy Configuration Program.
2. On the Licensing tab, click the Execute Key File button.
3. Type the path and filename of the **.skf** file in the Key file field, or click the Browse button and select the file.
4. Click the Execute button. The **lmu** utility executes the key file and installs all keys in the file for the current machine. See the "lmu Error Messages" table on page 2-38 for a list of possible errors returned by **lmu**.

You can also run the Synergy key file by double-clicking it from Explorer, or you can run **lmu** from the command line and specify the key file with the **-f** option. See "The lmu Utility" on page 2-28 for more information about **lmu** command line options.

If you do not have a Synergy key file

1. Start the Synergy Configuration Program.
2. On the Licensing tab, click the Install Keys Manually button.
3. Complete the fields in the Install Keys Manually dialog box:

   Product code. Select the product code from the drop-down list or, if the code you need is not listed, type it in.

   Producer code. Enter the code for the company that produces the product you're configuring. Synergex's producer code is 999.

   Configuration key. Enter the configuration key that Synergex sent you for this product. This key must be entered correctly! Pay special attention to occurrences of the number one (1) and the lowercase letter "L", as well as occurrences of the number zero (0) and the uppercase letter "O".
4. If this product requires an extended data string, click the Extended Data String button and enter it in the dialog box that displays. (This string is required only in special circumstances and will be given to you by Synergex.)
5. Click the Install Key button.
6. Repeat steps 3 through 5 to install the remainder of the keys. When you are through, click the Close button, and then click Apply on the Licensing tab.

Changing license configurations

Do not remove or edit registry entries or uninstall licensing to change a license configuration.

When you install Synergy/DE Client, if the machine is not already a license client, the license configuration will be changed to network license client and will use the network license server specified in the **setup.ini** file. See the installation instructions that accompanied your distribution for more information.

To change the licensee name on the server, see "Licensee name entered during installation is wrong" on page 2-16.

Changing a license client to use a different license server or port

1. Start the Synergy Configuration Program.
2. Click the Advanced button on the Licensing tab.
3. If the license server has changed, in the Advanced License Manager dialog, change the value in the Server name field. The server must have the necessary licensing in place before you can run applications on the client.
4. If the port for the license server has changed, or if the new license server is not running on the default port (2380), specify the port by adding a colon followed by the port number after the server name (e.g., Tiger:5555).
5. Click OK.

Changing a license client to a license server

You must have administrator privileges to change the license configuration from client to server.

1. Contact Synergy/DE Developer Support to get the current **-k** password.
2. From the command line run **lmu -k**, supplying the password as an argument.
3. Start the Synergy Configuration Program.
4. When you are prompted to configure licensing, select "License Server or Stand-alone". In the Unique Licensee Name field, enter the name that you provided to Synergex when you ordered Synergy/DE products for this machine.

Changing a license server to a license client

⚠ If you change a configuration from license server to license client, you cannot use the Synergy Configuration Program to change it back. See "Changing a license client to a license server" on page 2-8.

1. Start the Synergy Configuration Program.
2. Click the Advanced button on the Licensing tab.
3. In the Advanced License Manager dialog box, select Client and type the IP address or DNS name of the network license server in the Server name field. If your configuration includes a backup license server, you must use the DNS name. The server must have the necessary licensing in place before you can run applications on the client.
4. If the license server is not running on the default port (2380), specify the port by adding a colon followed by the port number after the server name (e.g., Tiger:5555).
5. Click OK.

Changing a license server's port

By default, License Manager server (**synd**) runs on port 2380. We recommend that you use this port if at all possible, because a non-default port must be specified not only on the license server, but—for network licensing—on *all* the license clients as well.

To change the License Manager server port, you will first need to change it from the command line on the license server (it cannot be changed from the Synergy Configuration Program on the server). Then you will need to specify the new port on each client, which can be done from either the Synergy Configuration Program or from the command line.

1. Change the port on the server using **lmu**:

   Reinitialize License Manager using the **-ns**, **-r**, and **-po** options, specifying the new port. Valid ports are in the range 1024 through 32767. For example,

   ```
   lmu -ns -r -po5555
   ```

   You'll be prompted to confirm the change. If you have stand-alone licensing, you are done.

2. Change the port on *each* license client from either the Synergy Configuration Program or the command line.

   - **Synergy Configuration Program:** When you start the Synergy Configuration Program, an error will display because the license server is not on the expected port. Click OK, and then click the Advanced button. In the Server name field, add a colon followed by the new port number to the end of the server name. For example, Tiger:5555.

- **Command line:** Reinitialize License Manager using the **lmu -nc** and **-r** options. Append a colon followed by the new port number to the end of the server name. You'll be prompted to confirm the change. For example,

  ```
  lmu -nc -rTiger:5555
  ```

You can specify the port number with the **-po####** option instead of appending it to the end of the server name. If both methods are used, the port specified with **-po** takes precedence.

Changing back to the default port

To change the license server back to the default port, do the following:

- Reinitialize License Manager using the **-ns** and **-r** options, but don't specify a port. For example,

  ```
  lmu -ns -r
  ```

  You'll be prompted to confirm the change.

To change the license clients back to the default, do one of the following:

- From the Synergy Configuration Program, click the Advanced button on the Licensing tab. Delete the colon and port in the Server name field and click OK.
- Reinitialize License Manager using the **-nc** and **-r** options, but don't append a port to the license server name or include the **-po** option. For example,

  ```
  lmu -nc -rTiger
  ```

Starting and stopping License Manager server (synd)

The License Manager server, **synd**, is a background process that runs on license servers and on machines that have local licensing. **Synd** runs as the service **synlm**. See "The synd Program" on page 2-27 for **synd** syntax and command-line options.

To start or stop License Manager server, you must have administrator privileges.

Starting License Manager server

License Manager server is started automatically when you install Synergy/DE core components.

Starting License Manager server manually

Do either of the following to start the service **synlm,** which starts **synd**:

- In the Windows Services console, select Synergy/DE License Manager and click the Start button.
- At a command prompt enter

  ```
  net start synlm
  ```

If you receive the error “Could not start the Synergy/DE License Manager service. The system could not find the file specified”, you should unregister and reregister the **synlm** service. See “The synd Program” on page 2-27 for the **synd** syntax. After reregistering, try starting **synlm** again.

The License Manager service **synlm** always logs its version and start/stop line to the Windows application event log. You can view the event log with the Windows Event Viewer. For more information on extended event logging for the **synlm** service, see “Error logging” on page 2-15.

Stopping License Manager server

Do any of the following to stop (but not unregister) the **synlm** service:

- In the Windows Services console, select Synergy/DE License Manager and click the Stop button.
- At a command prompt enter

```
synd -q
```

- At a command prompt enter

```
net stop synlm
```

To *stop and unregister* the **synlm** service, type the command **synd -x** at a command prompt.

Configuring a backup license server

A backup license server will take over the task of serving Synergy licenses in the event that the primary server becomes unavailable. Both the primary server and the backup server must be running Synergy/DE version 10 or higher. The license clients may be running older versions of Synergy/DE.

If the primary server goes down, the license clients will automatically switch to using the backup license server. The backup server can run for up to 21 days. You must restore the primary server (or designate a new primary server) before 21 days have elapsed in order to continue serving licenses. Once the primary server is restored, the serving of licenses automatically switches back to that machine.

The primary server checks continuously to ensure that the backup server is present. If the primary cannot detect the backup for a period of 24 hours, it disconnects itself from the backup and propagates the change to all the clients.

Requirements and recommendations

To implement a backup license server you must first purchase a Backup License Server (BLS) license. Contact your Synergy/DE account manager for information.

On the backup server machine, Synergy/DE must be installed to run locally, and the backup server must be a license client to the primary server. All clients, including the backup license server, must address the primary server by name, not by IP address.

We recommend the following:

- Use separate machines for your primary license server and your domain controller. If you do use a single machine for both purposes, you should have a backup domain controller in addition to a backup license server.
- Dedicate the primary server to the task of serving licenses. Synergy/DE applications and processes running on the primary server (including xfServer and xfServerPlus) will not fail over to the backup license server in the event the License Manager service on the primary server shuts down.
- Choose a single machine to serve as a backup license server. Although you can move the backup license server to another machine, or reconfigure the existing backup license server, you should do this only when absolutely necessary.

Each time you configure a license client to be a backup license server, the BLS license configuration counter is reduced. You'll see a warning message on the eighth configuration, which is the maximum. When you see this message, contact your Synergy/DE customer service representative to obtain a replacement configuration key for your BLS license. When you get the replacement key, configure it as described in "Installing configuration keys" on page 2-7. To check how many configurations you have remaining and how many replacement keys have been installed, see figure 2-3 on page 2-35.

Setting up the license servers

On the primary license server

1. Install Synergy/DE. When the installation program prompts you for the license configuration, select Network Server.
2. Configure license keys. You must configure the Backup Server License key (BLS10) on the primary server; this license key enables another machine to become a backup server to this primary server.
3. If you plan to run the primary license server on a non-default port, *change it now*, before setting up the backup license server. See "Changing a license server's port" on page 2-9.

On the backup license server

1. Install Synergy/DE to run locally. You must install at least core components.
2. When the installation program prompts you for the license configuration, select Network Client and specify the primary server by DNS name—not IP address—in the Server Name field.

   If the primary server is not running on the default port (2380), specify its port by appending a colon followed by the port number to the end of the server name (e.g., Tiger:5555).

⚠ If the backup server machine had a previous version of Synergy/DE installed, you will not be prompted for the license configuration during installation. Take a moment to verify that the backup server machine is a license client to the primary server. If it is not, stop! Change the license configuration of the backup server *now*, before continuing with this procedure. See “Changing license configurations” on page 2-8.

---

3. After the installation is complete, start the Synergy Configuration Program and click the Advanced button on the Licensing tab.
4. In the Advanced License Manager dialog box, select the “Be a backup server” option.
5. Click OK. You’ll see the backup server status displayed on the Licensing tab.

   Once the backup license server is configured, you will be able to see its status on the Licensing tab of the primary server or any of the license clients. See “Checking the status of the license servers” on page 2-14.

💡 By default, the backup license server will use the default port 2380 when it becomes the active license server. (This is the case even if the primary server was running on a non-default port.) Once the backup license server is configured, you can specify a different port using the **lmu -xb** command followed by the **lmu -nb -po####** command. Note that you do not have to do anything on the license clients to accommodate this change; they will automatically be notified.

On the license clients

When you install Synergy/DE on the license clients, specify the primary server by DNS name—not IP address—in the Server Name field. If the license clients are Synergy/DE Client machines, specify the name of the primary server in the **setup.ini** file.

⚠ The name of the backup license server is sent to the license client when the client makes a license request to the primary server. This value is then stored in the registry on the client. This means the client must have made a successful connection to the primary server in order to be able to use the backup server.

If the backup license server name changes, you should log onto each client machine as a user with elevated privileges and run a Synergy application. This will ensure that the new backup license server name is properly registered on the license client for all users.

Checking the status of the license servers

Once you have the primary and backup license servers configured, you can check their status from any client by looking at the Licensing tab in the Synergy Configuration Program. Next to the name of the primary and backup servers, you'll see a status of active, ready, or inaccessible.

- *Active* indicates which machine is currently being used as the license server.
- *Ready* means that the backup server is up and ready to take over should the primary go down.
- *Inaccessible* means the server cannot be accessed. This could be due to a loss of network connectivity or because License Manager server has stopped, or because the machine itself has been shut down. If the primary goes down when the backup is in this state, the license clients will not have access to a license server and will be unable to run Synergy programs.

Changing the backup license server for a primary server

To change which machine is the backup license server to a particular primary license server,

1. Configure a new machine to be a backup license server following the instructions in "On the backup license server" on page 2-13.
2. If the new machine has a different name, log onto each license client and run a Synergy application so that the client is informed of the name change. (See the important note above.)

   Once a new backup server is configured, the old backup server will automatically cease to be a backup server. This may take up to 10 minutes because that is how often the backup server checks for changes to the configured licenses. If desired, you can stop and restart the old backup server to expedite the process.

Changing the primary server for a backup license server

Follow these instructions if you want to change the primary server machine and retain the existing backup license server as a backup to the new primary. These instructions also apply if the primary server has gone down and you need to designate a new machine to replace it while the backup license server is running.

1. Install Synergy/DE on the new primary server and configure the backup license server key. If the new primary server has the same name as the old primary server, this is all you need to do.
2. If the new primary server has a different name than the old server, do the following on the *backup* server:
   - Start the Synergy Configuration Program and click the Advanced button on the Licensing tab.
   - Clear the "Be a backup server" check box and click OK. The Licensing tab redisplays.
   - Click the Advanced button again.
   - Change the value in the Server name field to the name of the new primary server and reselect the "Be a backup server" option.
   - Click OK in the Advanced License Manager dialog and then again on the Licensing tab.

Removing a backup license server

Follow these instructions to completely remove a backup license configuration. This procedure is done on the *backup* server. If you do not have access to that machine, see the note below for the command line procedure.

1. On the backup server, start the Synergy Configuration Program and click the Advanced button on the Licensing tab.
2. Clear the “Be a backup server” check box and click OK in the Advanced License Manager dialog.
3. Click OK on the Licensing tab. The backup license server will revert to client status and the change will be propagated to the primary server and all the license clients.

To remove a backup license server from the command line, run **lmu** with the **-xb** option. Normally, you would do this from the backup server machine; however, if that machine is not available or no longer exists, you can run **lmu -xb** from the primary server. It will remove backup licensing from the server and propagate the change to all license clients, including the backup license server.

Error logging

By default, the License Manager service **synlm** always logs its version, start-up and shut-down messages, and any errors it reports to the Windows application event log. You can view the event log with the Windows Event Viewer.

You can request more verbose logging in a separate log file. Verbose logging includes additional information such as licensing activity (logins and logouts) and the session and seat IDs. You can run **lmu** with the **-s** option to see these IDs, so that you can match them up with the logging output.

Synd buffers the logging information and then periodically writes it to the log, so unless there is a lot of licensing activity, you may not see anything in the log right away. You can force **synd** to write to the log by stopping the service.

To request verbose logging,

1. In the Synergy Configuration Program, click the Advanced button on the Licensing tab.
2. In the Advanced License Manager dialog box, select Enable verbose logging.
3. Change the default log file name and location, if desired. Include the complete path. If the log file specified does not exist, it will be created, but the directory path must exist. By default, the log is named **synd.log** and is located in the synergyde\dbl\bin directory.
4. Click OK.

Remember to review and clear the log from time to time. Do not leave verbose logging enabled indefinitely because the file will get quite large.

Uninstalling License Manager

License Manager is uninstalled when you uninstall Synergy/DE. Uninstalling removes only the License Manager files; it does not remove the license configurations. If you need to remove your license configurations, use the **lmu -k** option *before* uninstalling Synergy/DE. The **-k** option requires a password. Contact Synergy/DE Developer Support for the current **-k** password.

Troubleshooting License Manager

TIP You can use **synxfpng** with the **-s** option to troubleshoot License Manager network connection problems. See "The synxfpng Utility" on page 3-58 for details.

Because License Manager server uses TCP/IP to communicate, you may experience TCP/IP errors. See "Network Requirements (TCP/IP) for Client/Server Products" on page 1-10 for more information about TCP/IP.

Licensee name entered during installation is wrong

If you enter the wrong licensee name during installation, it will not match the licensee name given to Synergex when the license was purchased, and you cannot configure License Manager to run your Synergy/DE products. *Prior to configuring any products on the machine,* you can run **lmu** at the command line to change the licensee name. (You cannot change the licensee name from the Synergy Configuration Program.) You must have administrator privileges to run this command because it restarts License Manager server.

To change the licensee name, enter

```
lmu -rNewLicenseeName -ns
```

You'll be prompted to confirm the change.

If a Synergy Key file was generated with the *incorrect* name, you will need to regenerate it.

If products have already been configured, you must use **lmu -k**. Contact Synergy/DE Developer Support for the **-k** password.

Cannot access Network License Server error

The error "Cannot access Network License Server" (or sometimes just "Cannot access license server") can occur on license clients, as well as on machines with local licensing. This error may occur when running a Synergy application or when configuring the license client with **lmu**.

The error may be accompanied by a system error number, such as 11004 (usually on a client) or 10061 (usually on a server). There may be additional text (such as the server name that the client is attempting to connect to) that offers a clue to the problem. You may also want to check the Windows application event log for additional information.

Do you have a firewall?

IP-based licensing (introduced in version 10) uses a port (2380 by default) to communicate. If there is a firewall (any firewall, including the built-in Windows firewall) between a license server and a license client, there will need to be a firewall exception rule for License Manager. This is the case even with local (stand-alone) licensing because the machine is, in effect, serving licenses to itself. If License Manager can't get through your firewall, you will see a "cannot access network license server" error.

For network licensing: If there is a firewall between your license server and license clients, it needs to permit inbound access for License Manager (**synd.exe**).

For local licensing: For 10.1.1c and higher, a rule permitting License Manager access through Windows Firewall should be applied automatically.

To verify the rule is enabled,

1. Select Windows Firewall from the Windows Control Panel.
2. Select Advanced Settings in the left pane, and then click Inbound Rules to display the list.
3. Look for a rule named "Synergy License Server". The port is set to "any" so that it will work even if License Manager is running on a non-default port.
4. If the rule is not there, go to a command prompt and run "lmuelevate". Then refresh the list of inbound rules and check again for the "Synergy License Server" rule.

Troubleshooting non-firewall "cannot access…" issues

If you know a firewall is not the issue, the steps below can help you troubleshoot this problem. For a license client start with step 1; for a machine with local licensing, start with step 3.

1. On the license client, run the Synergy Configuration Program. You will see an error message ("Unable to query License Manager…") when it starts up. Click OK. The program will open and the server name will display on the Licensing tab.
2. If the server displayed is not the correct license server for this client, or if the port displayed is not the correct port, click the Advanced button to display the Advanced License Manager dialog box, and enter the correct name/port in the Server name field. (The port is required only when the license server is not running on the default port. Append a colon, followed by the port number to the end of the server name, e.g., Tiger:5555.)

You must have administrator privileges to perform the following steps on the license server machine.

3. If the server (and port, if non-default) displayed *is* the correct license server for this client, attempt to start the Synergy Configuration Program on the license server machine. If **synlm** is either not registered or not started, it will be registered and started when the Synergy Configuration Program starts up, and then the license client should be able to access the server.

4. If attempting to run the Synergy Configuration Program results in an error (“Unable to query License Manager…”), click OK in the error dialog and then close the program.

5. Go to a command prompt and attempt to run any **synd** command, such as **synd -h**. If you see the standard operating system error for an unrecognized application or command, it means **synd** is no longer there.

6. Go to Control Panel > Programs and Features. Locate Synergy/DE on the list of installed programs, right click on it, and select Change from the context menu to start the wizard. In the Program Maintenance dialog, select Repair. This will reinstall missing files. When it finishes, start the Synergy Configuration Program on the server.

7. If you still cannot access the license server, contact Synergy/DE Developer Support for assistance.

Setting Up License Manager on UNIX

License Manager is a set of utilities that controls the use of Synergy/DE products. On UNIX, License Manager runs as the **synd** daemon.

License Manager can also be used to secure the applications that you create with Synergy/DE. When a product is secured with License Manager, it will not run unless License Manager is installed and has been configured to recognize the product. To build License Manager protection into your own applications, you'll need Synergy/DE Licensing Toolkit. For more information about this product, view the *Licensing Toolkit User's Guide* or contact your Synergy/DE account manager.

On UNIX, License Manager enables all connected users to share licenses. To use License Manager—and your Synergy/DE products—you'll need to complete these steps:

1. Install the Synergy/DE products; License Manager is included with Synergy DBL. See "Installing License Manager" on page 2-20.
2. Obtain license configuration keys from Synergex. See "Obtaining configuration keys" on page 2-20.
3. Configure License Manager (i.e., install the configuration keys) to allow your Synergy/DE products to continue running after the 14-day evaluation period. See "Installing configuration keys" on page 2-21.

When you *first* install Synergy/DE on a machine that has never before had it installed, you will have unlimited evaluation use of all products for 14 days. After 14 days, you can run only the products that you have configured. Once you configure a particular product, your use of that product is limited to the number of licenses you purchased. After the 14-day evaluation period expires, if you want to evaluate a product, you must obtain a special configuration key from Synergex.

Each time you access the runtime during the evaluation period, the Synergex copyright message will display, along with the licensee name and the number of days remaining in the evaluation period. After you configure the runtime, this message no longer display.

Installing License Manager

These instructions are for new installations. If you are updating an existing installation, refer to the installation instructions that accompanied your distribution. If you need to reinstall Synergy/DE licensing software, call Synergy/DE Developer Support for assistance.

1. Install your Synergy/DE products using the installation instructions included with your distribution.

   When you execute the installation script program, **install.sde**, License Manager is initialized, the file **register.lm** is created, and the License Manager daemon (**synd**) is started. (For more information on **synd**, see "Starting and stopping the License Manager daemon (synd)" on page 2-22 and "The synd Program" on page 2-27.)

2. When prompted for the licensee name, enter the *unique* name that you provided to Synergex when you ordered Synergy/DE products. This name will display in the copyright notice when you run licensed Synergy/DE products.

   TIP We recommend that your company establish a licensee naming standard. If your company installs Synergy/DE on multiple systems at one site, we recommend that you use the name of the company followed by the user or machine name (e.g., "ABC Computers, Mary Smith" or "ABC Computers, Tiger"). If your company installs Synergy/DE at multiple sites, we recommend that you specify the company and city or location (e.g., "ABC Computers, Gold River" or "ABC Computers, store 215").

3. The licensee name you enter is echoed on the screen, and you are asked to confirm that it is correct.

   - If you entered the name *correctly*, type **y**. The screen displays the licensee name, registration string, machine name, and system code. This information is also written to the file **register.lm**.
   - If you entered the name *incorrectly*, type **n** and re-enter it.

   You can also use **lmu -r** to correct the licensee name. See "The lmu Utility" on page 2-28.

Obtaining configuration keys

There are two ways to obtain configuration keys.

- **Online.** If you are a Synergy/DE Developer Support customer and your Resource Center logon includes Online Keys access, you can obtain configuration keys using the Online Key Generation program on the Synergex website. See below for instructions. Contact your Synergy/DE customer service representative for more information about Online Keys access.
- **Manually.** You can fax or mail your registration information to Synergex, and then receive your keys by e-mail or fax.

Obtaining keys online

To complete this procedure, you will need your Resource Center user name and password. The configuration keys are delivered in a Synergy Key file.

1. Locate the file **register.lm** in the synergyde/lm directory and print it.
2. Start your web browser and go to the Synergex website.
3. Log into the Resource Center, go to the Online keys section, and click on the Generate Configuration Keys link. Follow the instructions on screen to obtain configuration keys.
4. When prompted, download the Synergy key file (**.skf** file) or select to have it sent to you as an e-mail attachment. Save the key file to your hard drive. It contains configuration keys for the products you are licensing. This key file can be run from **lmu** and greatly simplifies the configuration process (see "Installing configuration keys" on page 2-21).

Obtaining keys manually

1. Locate the file **register.lm** in the synergyde/lm directory and print it.
2. Use the information in **register.lm** to complete the Synergy/DE Product License Agreement. Carefully transcribe the licensee name, registration string, and system code onto the license agreement. Accuracy is important! Pay attention to case, punctuation, and spaces in the registration string.
3. Mail or fax the agreement to Synergex. You can indicate on the agreement how you want to receive your keys. If you select e-mail, you will receive a Synergy key file (**.skf** file) as an e-mail attachment. Save the file to disk, and use it to install the keys (see next section).

Installing configuration keys

1. If you have not already done so, set your PATH environment variable on the command line or in the **.profile** file to include the License Manager directory. For example:

   ```
   PATH=/usr/synergyde/lm:$PATH ;export PATH
   ```

2. Ensure that the system date and time are correct, and that no Synergy applications are running.

If you have a Synergy key file

Run **lmu** with the **-f** option and include the path and filename of the Synergy key file on the command line:

lmu -f*filename*.skf

This command runs the **lmu** utility and installs keys for all products in the specified key file for the current machine. See the "lmu Error Messages" table on page 2-38 for a list of errors returned by **lmu**.

If you do not have a Synergy key file

1. Execute the configuration utility:

```
config_lm
```

2. Enter the following information when prompted. You must enter this information for *each* Synergy/DE product you want to license. If you get **lmu** errors, see "lmu Error Messages" table on page 2-38.

   Product Code. Enter the code for the product you're configuring. Synergex will give you this code. For example, the product code for Synergy/DE Professional Series is PSDE10.

   Producer Code. Enter the code for the company that produces the product you're configuring. Synergex's producer code is 999.

   Configuration Key. Enter the key that Synergex sent you for this product. This key must be entered correctly! Pay special attention to occurrences of the number one (1) and the lowercase letter "L", as well as occurrences of the number zero (0) and the uppercase letter "O".

> If you need to enter an extended data string, you must configure the key using the command line syntax rather than with config_lm. Include the **-x** option for the extended data string. For example,
>
> ```
> lmu -a PROD9 -p 999 -i 58W4-ZA2T-BO42-X4AP -x 1234
> ```

3. After each configuration, you'll be prompted to configure another license. Enter **y** to configure another product or **n** to exit the configuration program.

Starting and stopping the License Manager daemon (synd)

The License Manager daemon, **synd**, is a background process that must be running in order for Synergy/DE products to run. It provides licensing and SEND/RECV message services to Synergy/DE products. It is initially started when you execute **install.sde**. (For information about the **synd** command-line syntax and options, see "The synd Program" on page 2-27.)

Starting License Manager automatically

To start License Manager automatically when the system reboots, add the full path name of **synd** to your system start-up file (the run command file (**rc**)). This ensures that **synd** will restart automatically if your system goes down. Consult your UNIX system administrator manual for more information about starting multi-user daemons.

For example, on some machines you might add this line to the **/etc/rc** file:

```
/usr/synergyde/lm/synd
```

Starting License Manager manually

To start License Manager manually, restart **synd** without rebooting or reinstalling. For example, on some machines you might enter

```
/usr/synergyde/lm/synd
```

Stopping License Manager

To stop License Manager, you must be UID=0. Use this command:

```
synd -q
```

Error logging

By default, **synd** logs caught signals (15 “Software terminate”, 11 “Segmentation error”, and so forth) to the log file **/usr/lib/synd.log** and stamps each signal with the date it occurred. You may want to examine this file when you have problems running License Manager. (Note: “Software terminate” is a normal signal to receive; it usually occurs when you shut down the machine.)

The default location for **synd.log** is /usr/lib, but it can be located elsewhere if the environment variable SYNDLOG is set. See SYNDLOG in the “Environment Variables” chapter of *Environment Variables & System Options*.

For more verbose debug logging, use the **lmu -2** option.

- To start debug logging by License Manager enter

  ```
  lmu -2
  ```

  (This command also starts debug logging for Synergy message manager. For more information on the Synergy message manager, see “Messaging” in the “Welcome to Synergy DBL” chapter of the *Synergy DBL Language Reference Manual*.)

- To stop debug logging by License Manager, wait at least 60 seconds after executing **lmu -2**, and then enter

  ```
  lmu -1
  ```

  This command sends a reset signal to License Manager. It can also be used to help restart **synd** if it gets into a frozen state.

Issuing two **lmu -1** commands within 60 seconds of each other will *start* debug logging.

The log file can get quite large if you do not periodically clear it out. To clear it out and start with a fresh file without stopping **synd**, run this command, when logged in as root:

```
>/usr/lib/synd.log
```

Troubleshooting License Manager

This section includes information on several common errors. Synergy DBL and other Synergy/DE products can generate errors that pertain to License Manager. See the "Error Messages" chapter of *Synergy Tools* for error descriptions.

synd: Insufficient privilege message

This message occurs when **synd** is not owned by root, the set-user-ID bit is not set, or the correct permissions are not set. Follow these instructions to correct the problem.

1. Log in as root or run **su**.
2. Move to the Synergy DBL lm directory and display a list of the directory:

   ```
   ls -la
   ```

3. Locate **synd** in the displayed list, and verify that the owner is root. If it is not, change the owner to root:

   ```
   chown root synd
   ```

4. Verify that the permissions for **synd** are set to `rwsr-xr-x`. The 's' is the set-user-ID bit. If the 's' is missing, set it:

   ```
   chmod u+s synd
   ```

%DBL-F-NOLMD Cannot access Synergy License Manager message

This error occurs because either **synd** is not running or there are extra Synergy message queues. To correct this problem, do the following:

1. Check to see if **synd** is running:

   ```
   ps -ef |grep synd
   ```

   If **synd** is running, you'll see something like this:

   ```
   root 10343     1  0 15:49:51   ttyp0    00:00:00 synd
   ```

2. If **synd** is not running, check your PATH setting. It should reference the lm and dbl/bin directories, which should be located in the same parent directory. For example:

   ```
   /usr/synergyde/lm;/usr/synergyde/dbl/bin
   ```

   If PATH is incorrect, reset it.

3. Start **synd** manually:

   ```
   synd
   ```

4. Try to run your Synergy application again.
5. If the error still occurs, check again whether **synd** is running.

6. If **synd** is still not running, there may be a problem with the Synergy/DE message queues. Get the queue ID numbers:

   ```
   lmu -q
   ```

   This displays the two Synergy/DE message queues used by License Manager. The output will look similar to this:

   ```
   Queues: (ID=4 KEY=67113158<0x40010c6>) (ID=3 KEY=83890374<0x50010c6>)
   Synergy Daemon: (synd) pid 0
   ```

   If either ID is -1, **synd** was unable to create the message queues.

7. View all the queues on the system:

   ```
   ipcs -qa
   ```

8. Find the Synergy queues. The Synergy queues can be identified by the pair of keys that start with 0x20 and 0x30 (version 5) or 0x40 and 0x50 (version 6 and higher), followed by a number, which is the same for both keys. For example:

   ```
   0x40010c6
   0x50010c6
   ```

   Check the number of queues in use and the total number of bytes in the queues (in the CBYTES column). There may be extra Synergy queues that you do not need, especially if you've reinstalled. You need to either remove unnecessary queues or increase the number or size of the queues in the kernel. To increase the queue resources on your machine, check with your system administrator.

9. Be careful removing queues. Remove only queues that you are *certain* can be removed! If you are uncertain whether you have extras queues or which queues should be removed, stop now and call Synergy/DE Developer Support.

   To remove queues enter

   `ipcrm -q`*q_number*

   where *q_number* is the ID of the queue you want to remove. You'll do this twice—once for each queue ID in the pair.

10. Restart **synd** manually and try to run your Synergy application. If PATH, DBLDIR, and all permissions are set correctly, you should be able to run your application.

11. If the application still fails after these steps have been taken, call Synergy/DE Developer Support.

LMFAIL error followed by MSGWAIT error

These errors happen when **synd** takes too long to respond to either the Synergy runtime or to the **lmu** utility, and can occur when executing the **install.sde** and **config_lm** scripts. This can be caused by a slow system or network.

1. If a retry or two aren't successful, check to see if the **synd** daemon is running:

   ```
   ps -ef | grep synd
   ```

2. If the **synd** daemon is not running, restart **synd** manually, and then try to run your Synergy application again:

   ```
   synd
   ```

3. You can also set the MSGWAIT environment variable to wait longer for the communication before issuing the MSGWAIT error. The MSGWAIT environment variable should be used only as a temporary measure until the slow network or system slowdown is corrected. (For more information, see MSGWAIT in the "Environment Variables" chapter of *Environment Variables & System Options*.) If you have other applications using message queues, you may be exceeding the system's maximum number of message segments. Try increasing the kernel parameter MSGSEG. (This parameter may have a different name on different UNIX platforms.)

The synd Program

On Windows, License Manager server runs as the **synd** service, **synlm**. On UNIX, License Manager runs as the **synd** daemon.

Syntax

`synd` *[-option][...]*

-h

Display a list of **synd** options.

-v

Display the **synd** version.

Windows options

-q

Stop the **synlm** service.

-r

Register the **synlm** service.

-rs

Register the **synlm** service (if it is not already registered) and then start it.

-x

Unregister the **synlm** service. If the service is running, **-x** will stop it and then unregister it.

UNIX options

<none>

Start the License Manager daemon.

-b*size*

Change the message queue buffer size. The default is 4096. Change this value if you SEND/RECV messages larger than 4096 bytes.

-p

Ping the License Manager daemon to see if it is responding; also displays **synd** version.

-q

Stop the License Manager daemon. You must be UID=0 to do this.

The lmu Utility

The **lmu** utility is used to initialize or reinitialize License Manager, configure License Manager to allow Synergy/DE products to run, and generate license usage reports. See "Windows options" below or "UNIX options" on page 2-32.

Syntax

`lmu` *[-option][...]*

-h

Display a list of **lmu** options.

-v

Display the **lmu** version.

Windows options

<none>

Display concurrent usage and configuration information. See figure 2-1 on page 2-34. This command will first start License Manager, if it is not already running. To send the output to a file, use **-o**.

-app_code

Display license information for the specified *app_code*(s). You can list multiple application (product) codes on the command line. See also the **-u** option. See "Displaying license usage information" on page 2-36 for sample output.

-a*app_code*

Configure product *app_code*. Used with **-i** and **-p**.

-b

Brief display of license information, including licensee name, registration string, and system code. On a Windows license client, the license server name and port display instead of the licensee name. On a Windows license client or server, if a backup license server is configured, its server name and port will also display.

-c*lic_name*

Initialize License Manager on a license server for licensee *lic_name*.

The minimum number of characters for *lic_name* is 4; the maximum is 50. Valid characters are alphanumeric; if the name is longer than 4 characters, it may include spaces and punctuation characters. If *lic_name* contains spaces, it must be enclosed in quotation marks (" "). Used with **-ns**.

-c*svr_name[:port]*

Initialize License Manager on a license client for server *svr_name.*

Svr_name may be an IP address or the server DNS name. The minimum number of characters for *svr_name* is determined by your Windows operating system; the maximum is 255 (including the port, if specified). Valid characters are alphanumeric, period (.), and hyphen (-).

You may optionally specify a non-default port number that **synd** is running on. The default port is 2380. Valid ports are in the range 1024 through 32767. Used with **-nc**.

-d

Initialize License Manager (prompted).

-f*filename*

Configure License Manager using the Synergy key file *filename* (the **.skf** file). If *filename* is not in the current directory, specify the complete path. This command installs all the product keys in the **.skf** file for the machine from which it is run, and then displays a list of products for which keys were installed.

-i*config_key*

Specify configuration key *config_key.* Used with **-a** and **-p**.

-k*password*

Remove licensing with a *password* obtained from Synergy/DE Developer Support. This password is valid for seven days. After running the **-k** option, you'll need to reinitialize License Manager. Use **lmu -c** or **lmu -d** to reinitialize licensing and then reconfigure all products.

-l*lang_code*

Initialize or reinitialize License Manager with foreign language copyright messages. Valid codes:

1	Dutch
2	French
3	Italian

-m*filename*

Display the contents of the Synergy key file *filename* for the machine *on which the command is run*. This command may not display all data in the file. See also the **-t** option.

-nb

Initialize a license client as a backup server. To specify a non-default port, use **-po**. Before running this option, licensing (including a BLS license) must be configured on the primary server, and the backup server must be configured as a license client to the primary server. See "Configuring a backup license server" on page 2-11 for additional requirements.

-nc

Network client installation. Used with **-c** or **-r**.

-ns

Network server installation. Used with **-c** or **-r**.

-o*filename*

Send concurrent usage and configuration information to a file instead of the screen. This option applies only to the output that is produced when you run **lmu** without any options.

-p*id*

Specify producer ID *id*. Used with **-a**, **-i**, and **-u**. Required with **-u** when producer code is other than 999.

-po*port*

Specify a non-default port for the License Manager service. The default is 2380. Valid ports are in the range 1024 through 32767. Used with **-ns**, **-nc**, and -**nb**. This option is primarily for use with **-ns**, where the licensee name is specified instead of the server name. When you run **lmu** with the **-nc** option, you can specify the port as part of the server name (**-c** or **-r**); if you also specify **-po**, it is used instead. If **-po** is specified without a port number, the port is set to the default, 2380.

-r

Reinitialize License Manager and change the license server port. When used without an argument, **-r** can be used with **-ns** and **-po** to change the server to a non-default port. When used with just **-ns**, it changes the port back to the default. See “Changing a license server’s port” on page 2-9 and “Changing back to the default port” on page 2-10.

-r*lic_name*

Reinitialize License Manager on a license server where *lic_name* is the licensee name. Used with **-ns**. Use this option to change the licensee name for a license server prior to configuring products. If you have already configured products, you must use **lmu -k**. See the **-c***lic_name* option on page 2-28 for details on valid values for *lic_name*. This option can be used without specifying the licensee name to change the port on the server.

-r*svr_name[:port]*

Reinitialize License Manager on a license client where *svr_name* is the license server name. Used with **-nc**. Use this option to change the license server used by an existing license client or change a license server to a license client. Specify the port number if the license server is running on a non-default port. See the **-c***svr_name[:port]* option on page 2-29 for details on valid values for *svr_name* and *port*.

- To change the license server for an existing license client, use **-r***new_server_name* **-nc**.
- To change a license server to be a license client of server *svr_name*, use **-r***svr_name* **-nc**. If the machine was acting as a license server, all connections from its license clients will be lost. Note that you cannot use the **-r** option to change a license client back to a license server; you must use **lmu -k**. See "Changing a license client to a license server" on page 2-8 for details.

-s

Display the License Manager seat ID. Knowing the seat ID is useful when displaying detailed license usage or when tracking license log-ins using verbose logging (see "Error logging" on page 2-15), because you can match up the ID with the usage or debug output. On a terminal server machine, this option also displays details such as whether the workstation is a client or console and whether install mode is enabled.

-t*filename*

Display the contents of the Synergy key file *filename*, sorted by registration string. This option displays all data in the file, regardless of whether it is for the current machine. See also the **-m** option.

-u*app_code*

Display detailed license usage for the specified *app_code*(s). You can list multiple application (product) codes on the command line, separated by a space. Specify **-p** with **-u** when the producer ID is other than 999. See "Displaying license usage information" on page 2-36 for sample output.

-x*string*

Specify extended data string; used only with special application codes.

-xb

Remove backup licensing. This option is normally run from the backup server, but it can be run from the primary server if the backup server no longer exists or is not accessible. It cannot be used with any other option. Once **-xb** is run, the backup server reverts to license client status and the change is propagated to all license clients. Do not use this option to change either the backup license server or the primary server; instead, follow the instructions in "Changing the backup license server for a primary server" on page 2-14 and "Changing the primary server for a backup license server" on page 2-14.

UNIX options

<none>

Display concurrent usage and configuration information. See figure 2-1 on page 2-34. To send the output to a file, use **-o**.

-app_code

Display license information for the specified *app_code*(s). You can list multiple application (product) codes on the command line. See also the **-u** option. See "Displaying license usage information" on page 2-36 for sample output.

-a*app_code*

Configure product *app_code*. Used with **-i** and **-p**.

-b

Brief display of license information, including licensee name, registration string, and system code.

-c*lic_name*

Initialize License Manager for licensee *lic_name.* The minimum number of characters for *lic_name* is 4; the maximum is 50. Valid characters are alphanumeric; if the name is longer than 4 characters, it may include spaces and punctuation characters. If *lic_name* contains spaces, it must be enclosed in quotation marks (" ").

-f*filename*

Configure License Manager using the Synergy key file *filename* (the **.skf** file). If *filename* is not in the current directory, specify the complete path. This command installs all the product keys in the **.skf** file for the machine from which it is run, and then displays a list of products for which keys were installed.

-i*config_key*

Specify configuration key *config_key.* Used with **-a** and **-p**.

-k*password*

Remove licensing with a *password* obtained from Synergy/DE Developer Support. This password is valid for seven days. After running the **-k** option, you'll need to reinitialize License Manager. Use **install_lm** to reinitialize; contact Synergy/DE Developer Support for information about **install_lm**.

-l*lang_code*

Initialize or reinitialize License Manager with foreign language copyright messages. Valid codes:

1	Dutch
2	French
3	Italian

-m*filename*

Display the contents of the Synergy key file *filename* for the machine *on which the command is run*. This command may not display all data in the file. See also the **-t** option.

-o*filename*

Send concurrent usage and configuration information to a file instead of the screen. This option applies only to the output that is produced when you run **lmu** without any options.

-p*id*

Specify producer ID *id*. Used with **-a**, **-i**, and **-u**. Required with **-u** when producer code is other than 999.

-q

Display Synergy/DE message queue information.

-r*lic_name*

Reinitialize License Manager for licensee *lic_name* prior to configuring products. See the **-c** option above for details on valid values for *lic_name*.

-t*filename*

Display the contents of the Synergy key file *filename*, sorted by registration string. This option displays all data in the file, regardless of whether it is for the current machine. See also the **-m** option.

-u*app_code*

Display detailed license usage for the specified *app_code*(s). You can list multiple application (product) codes on the command line, separated by a space. Specify **-p** with **-u** when the producer ID is other than 999. See "Displaying license usage information" on page 2-36 for sample output.

-x*string*

Specify extended data string; used only with special application codes.

-1

Stop debug logging and send a reset signal to License Manager. See "Error logging" on page 2-23 for more information.

-2

Start debug logging by License Manager. See "Error logging" on page 2-23 for more information.

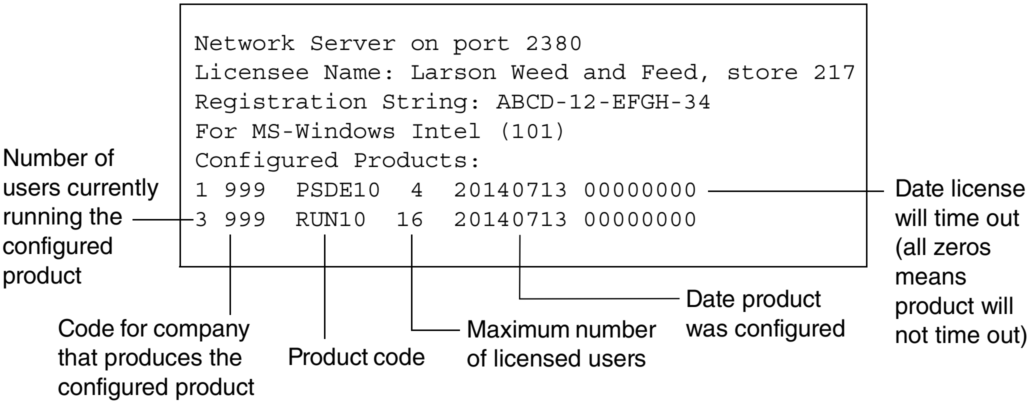
Discussion

The **lmu** utility must be run from the synergyde\dbl\bin directory if the bin directory is not in your path.

See the “lmu Error Messages” table on page 2-38 for a list of error messages returned by the **lmu** utility and assistance in resolving them.

Displaying license configuration information

Running the **lmu** utility without any arguments displays to the screen information about license configuration, including the registration string, product code, and current and maximum number of users. On a Windows license server, you’ll see the port number that it is running on. On a Windows license client, you’ll see “Network Client with Server *name* on port ####”, along with the products that are configured on the license server. Figure 2-1 shows a sample report on a Windows license server. You can run **lmu** with the **-o** option to send this information to a file.



```
Network Server on port 2380
Licensee Name: Larson Weed and Feed, store 217
Registration String: ABCD-12-EFGH-34
For MS-Windows Intel (101)
Configured Products:
1 999  PSDE10  4  20140713 00000000
3 999  RUN10  16  20140713 00000000
```

Number of users currently running the configured product

Date license will time out (all zeros means product will not time out)

Code for company that produces the configured product

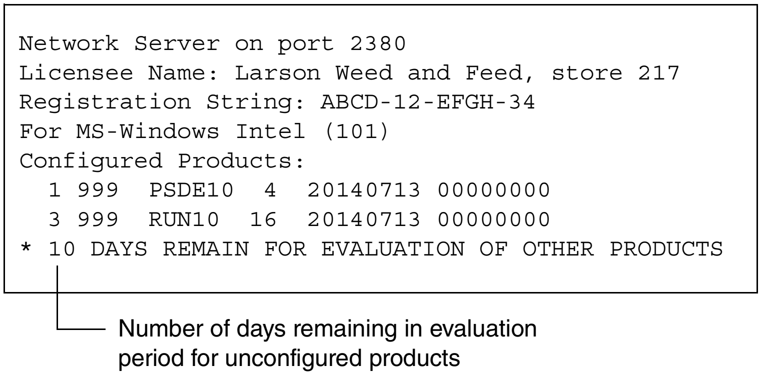
Product code

Maximum number of licensed users

Date product was configured

Figure 2-1. Sample lmu report showing configured products for a license server.

If the 14-day evaluation period has not yet expired, you'll see a message telling you how many days remain for evaluation of the installed products. Note that some products may already be configured, while others are still running as evaluation copies, as shown in figure 2-2.

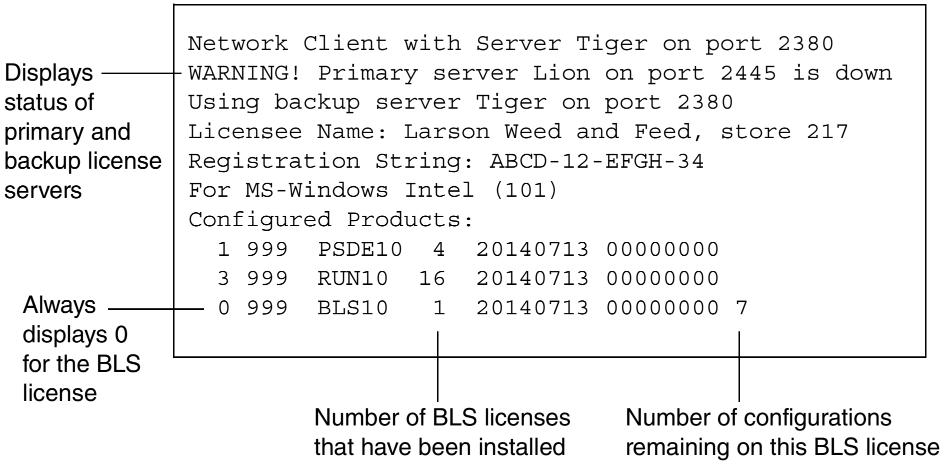


```
Network Server on port 2380
Licensee Name: Larson Weed and Feed, store 217
Registration String: ABCD-12-EFGH-34
For MS-Windows Intel (101)
Configured Products:
  1 999  PSDE10  4  20140713 00000000
  3 999  RUN10  16  20140713 00000000
* 10 DAYS REMAIN FOR EVALUATION OF OTHER PRODUCTS
```

Number of days remaining in evaluation period for unconfigured products

Figure 2-2. Sample lmu report showing evaluation period remaining.

If you are using a backup license server (Windows only), the **lmu** report will tell you how many BLS keys have been installed and the number of configurations remaining as shown in figure 2-3. The maximum number of configurations is 8; when the displayed value reaches 1, you should contact your Synergy/DE customer service representative to obtain a replacement configuration key. The status of the primary and backup license servers is also displayed.



```
Network Client with Server Tiger on port 2380
WARNING! Primary server Lion on port 2445 is down
Using backup server Tiger on port 2380
Licensee Name: Larson Weed and Feed, store 217
Registration String: ABCD-12-EFGH-34
For MS-Windows Intel (101)
Configured Products:
  1 999  PSDE10  4  20140713 00000000
  3 999  RUN10  16  20140713 00000000
  0 999  BLS10   1  20140713 00000000 7
```

Displays status of primary and backup license servers
Always displays 0 for the BLS license
Number of BLS licenses that have been installed
Number of configurations remaining on this BLS license

Figure 2-3. Sample lmu report showing backup license server information from a license client. The backup server is currently in use.

Displaying license usage information

Running **lmu** and specifying an application (product) code displays license usage information for the specified product. You can include multiple application codes on the command line by separating them with a space. The example below shows output for the runtime (RUN10) ad Workbench (PSW10). Each product is configured for 10 users. There are currently two users using runtime licenses and one user using a Workbench license.

```
c:\lmu RUN10 PSW10
License: RUN10 Producer: 999    (10 User)
  Configured on July 13, 2014
  Current usage: 2

License: PSW10 Producer: 999    (10 User)
  Configured on July 13, 2014
  Current usage: 1
```

To see more detail, run **lmu** with the **-u** option followed by the application code(s). On the Windows sample below, the seats match the seat IDs that display when you run **lmu -s**. This will tell you exactly who is using the licenses. On UNIX, the seats match the process IDs of the applications taking the licenses. You can look at the processes on the system to match the seat with a user name.

```
c:\lmu -u RUN10 PSW10
License: RUN10 Producer: 999    (10 User)
  Configured on July 13, 2014
  Current usage: 2
      Seat 1: MICKEYf75dc5f1 - 1 active process
      Seat 2: BARNEYbe71aa31 - 1 active process (Non-interactive)

License: PSW10 Producer: 999    (10 User)
  Configured on July 13, 2014
  Current usage: 1
      Seat 1: MICKEYf75dc5f1 - 1 active process
```

Displaying Synergy key file contents

Running **lmu** with the **-t** option displays the contents of a Synergy key file (**.skf**). The output includes the registration string, licensee name, and a list of the products that can be configured from the file. There may be data for more than one licensee (i.e., more than one machine) in the file, as shown in the example below.

```
c:\lmu -t 000999_SY101091605.skf
ABCD-12-EFGH-34 - Larson Weed and Feed, store 217
        PSDE10            Professional Series Development Environment
        RUN10             Synergy Runtime
IJKL-67-MNOP-93 - Larson Weed and Feed, store 354
        RUN10             Synergy Runtime
        PSW10             Professional Series Workbench
```

To see the keys only for the system on which you are running the command, use the **-m** option. The output will look similar to the following:

```
c:\lmu -m 000999_SY101091605.skf
The specified Synergy Key file contains the following keys for this
system:
       PSDE10      Professional Series Development Environment
       RUN10       Synergy Runtime
```

Examples

See the Discussion above for examples of using **lmu** to display information about licensing.

Windows

- To initialize License Manager as a license client to server *Tiger* on the default port, use

  ```
  lmu -cTiger -nc
  ```

- To change the server name for a license client, run this command on the client. *Lion* is the new server name, which is running on the default port. (This command can also be run on a license server to change it to a license client.)

  ```
  lmu -rLion -nc
  ```

  If the new server is running on a non-default port, append the port number to the end of the server name:

  ```
  lmu -rLion:5555 -nc
  ```

- To initialize License Manager as a license server on a non-default port for licensee *Fred*, use

  ```
  lmu -cFred -ns -po5555
  ```

UNIX

- This example shows how to configure a license for the runtime (RUN10). This command includes the producer ID (999) and the configuration key.

  ```
  lmu -aRUN10 -p999 -i58W4-ZA2T-BO42-X4AP
  ```

Error messages returned by lmu

If you get errors while running **lmu**, check the table below to determine the cause.

lmu Error Messages

Message	**Cause**
Backup license server requires a client license	You attempted to configure a backup license server on a system that is not configured as a license client. The backup license server must be a license client of the primary license server. See “Changing license configurations” on page 2-8. (Windows)
Backup license server requires an active Primary Server.	You attempted to configure the backup license server when the License Manager server on the primary server was not available.
Backup server already configured. To make changes, first revert back to a client with (lmu -xb)	You issued an **lmu -nb** command on a machine already configured as a backup license server. Before making any changes to the backup server configuration (such as the port number), you must first “undo” it by running **lmu -xb**. See “Configuring a backup license server” on page 2-11. (Windows)
Backup server license is not configured	You must configure a backup license server key on the primary server before initializing the backup server. See “Configuring a backup license server” on page 2-11. (Windows)
Backup server not allowed from 32-bit client on 64-bit machine	A 32-bit license client running on a 64-bit machine cannot be a backup server because the 32-bit **synd** is not installed in this configuration. (Windows)
Cannot access license file	**Lmu** cannot access the license information. The file may not exist.
Cannot access license server	The License Manager server is not running on your license server machine on the specified port or the server name specified on the client is incorrect. See “Troubleshooting License Manager” on page 2-16. (Windows)
Cannot access newer license information	The version of License Manager you are running is incompatible with the license information. You need a newer version of **lmu**. (Windows)

lmu Error Messages (Continued)	
Message	**Cause**
Cannot access primary server	You attempted to configure a backup license server to a license server other than the primary server. (Windows)
Changing licensee name is not allowed once licenses have been configured lmu: Reinstall aborted	You ran **lmu -r -ns** after products had already been configured. See "Troubleshooting License Manager" on page 2-16.
Configuration limit for BLS exceeded	You have exceeded the limit for configuring the backup license server. You can configure a backup license server eight times for a given BLS license. Contact your Synergy/DE customer service representative for a replacement key. (Windows)
Corrupted license file	The license database is damaged. Contact Synergy/DE Developer Support.
Corrupted license information	The license database in the registry is damaged. Contact Synergy/DE Developer Support.
File name too long; max 255 characters allowed	The Synergy key file name you supplied is too long.
Insufficient privilege	You aren't running with sufficient privileges to restart **synd** with your configuration changes. Or, the set-user-ID bit is not set for **synd**. See "synd: Insufficient privilege message" on page 2-24. (UNIX)
Invalid configuration key	You either typed the configuration key incorrectly or incorrectly relayed your registration string to Synergex. Re-type the key, paying special attention to occurrences of the number one (1) and the letter "l" and occurrences of the number zero (0) and the letter "O". If you still get this message, contact Synergex to see if you provided an incorrect registration string. (To see the registration string, use the **-b** option.)
Invalid language code	You entered an invalid language code. See page 2-29 for the valid codes.
Invalid network server port number	The port specified is not in the range 1024 through 32767. (Windows)

lmu Error Messages (Continued)	
Message	**Cause**
Invalid option (**-n*x***): network type already specified	Where *x* is **b**, **c**, or **s**. You've entered two network types (backup server, client, or server) on the command line. (Windows)
Invalid registration string	The license file is not registered for this machine. Contact Synergy/DE Developer Support.
Invalid use of IP address *address*	The backup license server requires that the primary server be specified by DNS name rather than IP address.
License file already installed	You have already initialized the licensing system.
License installation corrupt	Your license registry has been changed in error or removed. Use **lmu -k**. Contact Synergy/DE Developer Support for the password and assistance.
License manager did not start… License manager did not re-start… License manager did not stop… service error	The license service could not be started, restarted, or stopped. (Windows)
License server install requires a local drive image	You attempted to configure a Synergy Client machine as a license server. To be a license server, the machine must have Synergy/DE installed to run locally. (Windows)
Missing language code	You didn't enter a language code.
Missing required application code	You didn't type anything at the Product Code prompt.
Missing required configuration key	You didn't specify the **-i** (configuration key) option or you entered a blank configuration key.
Missing required licensee name	You didn't type anything at the Licensee Name prompt.
Missing required network type	You didn't specify **-nc** or **-ns** to indicate network client or network server. (Windows)
Missing required producer ID	You didn't type anything at the Producer Code prompt.
Network server name too long	The name of the network server exceeds the limit of 255 characters. (Windows)

lmu Error Messages (Continued)

Message	**Cause**
No Synergy Key Files specified	You ran **lmu** with the **-f**, **-m**, or **-t** option but didn't supply a Synergy key file name.
Old version of Synergy License Manager	You did not upgrade your License Manager software when you upgraded some other component of Synergy/DE. Install the latest version of the License Manager software.
Producer file not found	You entered an invalid response at the Producer Code prompt.
Products can only be configured on server	You attempted to install configuration keys on a license client. In a client/server configuration, configuration keys are installed only on the license server. (Windows)
Primary server has no configured products.	No products, other than BLS, have been configured on the primary license server. (Windows)
Server port already in use	The port you have specified for the license server is already in use; choose a different port. (Windows)
Synergy Key File is invalid	The specified Synergy key file is invalid (e.g., contains a misspelled keyword or is missing a required keyword). Contact Synergy/DE Developer Support.
The specified Synergy Key File does not contain any keys for this system	The filename you specified with the **-f** or **-m** option doesn't contain any keys for the current system.
Uninitialized license file	Your license installation has become corrupt. Use **lmu -k**. Contact Synergy/DE Developer Support for the password and assistance.
Wrong configuration key for installation	The configuration key is not compatible with your installation. For example, the registration string or the licensee name may be wrong. Verify that the configuration key you entered is intended for the current system.
Wrong product code	You entered an invalid product code or a configuration key from another product configuration.
You must install the license file first	You attempted to configure licenses before installing License Manager.

3

Configuring xfServer

Getting Started with xfServer

Before you begin, we recommend you read the “Accessing Data Remotely with xfServer” chapter in *Getting Started with Synergy/DE* for a general description of xfServer and how to use it.

There are four steps that you should take in preparation for using xfServer:

1. Install your client and server systems

- On your server system, follow the instructions that came with your Synergy/DE distribution to install xfServer.
- On your client systems, install Synergy DBL 6.1 or higher. (Version 9.3 or higher is required for encryption.) Client capabilities are included in the runtime. The client version must be the same as the server or lower. Using a newer client with an older server is not a supported configuration and causes the error “Unsupported server version/Feature not available” (SRVNOTSUP).
- Review the Synergy DBL and xfServer release note files (**REL_DBL.TXT** and **REL_SRV.TXT**) for any additional features or last-minute information.

2. Ensure TCP/IP communication

Use the **synxfpng** utility to ensure that your client and server systems are able to communicate via TCP/IP. See “The synxfpng Utility” on page 3-58.

3. Organize your data files

Make sure the data files you want the clients to share are located on the server and have the correct permissions.

4. Prepare your application

To prepare your application, do the following:

- Determine the environment variables you will use to specify the server name or address. See “Why use environment variables with xfServer?” on page 3-4 for information on the advantages of using environment variables.
- Use those environment variables within your application’s file specifications to specify file locations. Note that a remote file specification cannot reference another remote file specification when the first file specification is on Windows.
- Define the environment variables on the client to point to the correct server (or local) address.
- Decide which, if any, routines need to be encrypted and make the necessary adjustments to your code. See “Using Client/Server Encryption” on page 3-7.

Restrictions

Note the restrictions for the following statements and subroutines when used with xfServer.

- COPY subroutine—Supported only for text and binary files on OpenVMS xfServer. Copying ISAM files to and from OpenVMS is not supported.
- GETFA subroutine—Only the following keywords can be used with xfServer: IDK, IKL, IKP, ISZ, KRF, NDN, NKY, OMD, ORG, OST, RFA, RMT, RSZ, VER. All other keywords require a local file system and cannot use xfServer file syntax.
- LPQUE statement—Supports printing local files on printers accessible to the local machine and printing remote files on printers accessible to the remote machine. On Windows, we recommend that the client generate a print file locally and then print it on a printer accessible to the client. We do not recommend that the client print a remote file using xfServer; however, if you choose this configuration, the printer must be attached directly to the server machine. Network printers are not supported from xfServer on Windows.
- MERGE statement—Not implemented on OpenVMS xfServer.
- RENAM subroutine—When using remote specifications for *new_name* and *old_name*, any server name that is specified as part of *new_name* is ignored, and *new_name* is assumed to be on the same machine as *old_name*.
- SETDFN subroutine—Not implemented on xfServer.
- SORT statement—A temporary file is created by SORT and placed in the following default locations:
  - On Windows, it is placed in the directory that TMP or TEMP is set to in the Windows registry (not the environment). Use the Synergy Configuration Program to specify a location; see “Defining environment variables for xfServer services” on page 3-25. If TMP/TEMP are not set, the file is placed in c:\Users\Public\Public Documents.
  - On UNIX, it is placed in /tmp.
  - On OpenVMS, it is placed in SYS$SCRATCH.

  See SORT in the “Synergy DBL Statements” chapter of the *Synergy DBL Language Reference Manual* for more information about remote sorting.

Why use environment variables with xfServer?

Although you can hard-code server specifications in your program, to keep your xfServer system portable and manageable you must define environment variables and use them in your application's filename specifications. If you hard-code the entire file specification in your program, and those files move, or if your user doesn't set up the system exactly as you expect, you will need to change the file specifications and recompile your program.

For example, instead of using a hard-coded method such as this for file access:

```
open(1,u:i,"filename@server")
```

We recommend using an *environment variable* assignment (or *logical* on OpenVMS), such as this:

```
open(1,u:i,"env_variable:filename")
```

The environment variable replaces the server name in your actual code.

The following examples show two ways that you might define the DAT environment variable:

- On Windows and UNIX xfServer, to point to the default directory, as defined by RSFILPATH set on the server, use

  ```
  DAT=@server_name
  ```

  On OpenVMS, this will point to the user's home directory. (See RSFILPATH in the "Environment Variables" chapter of *Environment Variables & System Options* for more information about this environment variable.)

- To point to the SRV environment variable, as defined by xfServer, use

  ```
  DAT=SRV:@server_name
  ```

  In this case, all users go to the same directory. You must also define the SRV environment variable on the xfServer machine. For instructions on defining environment variables for the xfServer service, refer to the following operating system-specific information:

  - Windows, see "Defining environment variables for xfServer services" on page 3-25
  - UNIX, see "Defining environment variables for xfServer processes" on page 3-30
  - OpenVMS, see "Defining logical names for xfServer processes" on page 3-36

Let's say that you've defined `DAT=SRV:@server`, and your application has the statement:

```
open(1,u:i,"DAT:cusmas")
```

The `DAT:cusmas` data request might be translated as shown in figure 3-1 for a Windows client connected to a UNIX server.

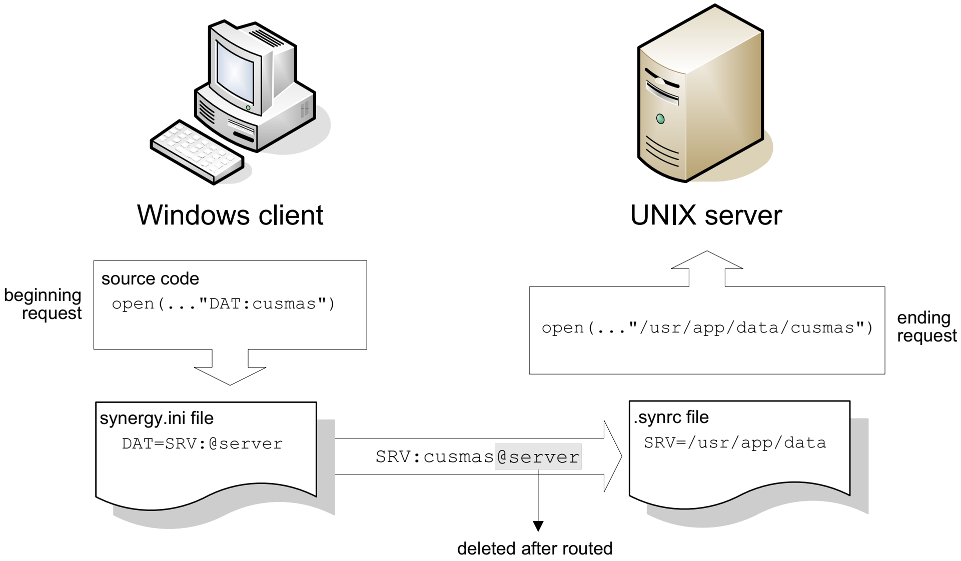


Figure 3-1. Translating a request from a Windows client to a UNIX server using an environment variable.

Prefetching and Buffering Records

You can improve xfServer performance by prefetching and buffering records.

The prefetch feature improves sequential READS performance. This feature enables the client to prefetch sequential records from the server and store them in a buffer until they are needed. Prefetching is used with files of any type that are opened in input mode, with relative files that are opened in update mode, or with ISAM files that are opened in update mode and use the LOCK:Q_NO_TLOCK option.

TIP If you currently have Q_NO_LOCK set and wish to use prefetching, you can set system option #55. This automatically maps READS with LOCK:Q_NO_LOCK to READS with LOCK:Q_NO_TLOCK, so that you can take advantage of prefetching without having to alter your code. See system option #55 in the "System Options" chapter of *Environment Variables & System Options* for details and restrictions.

Buffering records improves sequential WRITES and PUTS performance. This feature enables the client to buffer sequential records waiting to be written to the server and then write them all at once, which improves file loading performance. Buffering is used with relative and sequential files that are opened in output or append mode (WRITES), and with stream files that are opened in output or append mode (WRITES or PUTS). Buffering does not take place when WRITES is used with the GETRFA option.

Both prefetching and buffering are controlled by the SCSPREFETCH environment variable. By default, prefetching/buffering is off. To turn these features on, set SCSPREFETCH on each client machine to a value between 1 and 32. This indicates the size of the buffer in kilobytes (e.g., a value of 4 results in a buffer size of 4096 bytes). Setting SCSPREFETCH to 0 turns off prefetching/buffering. See SCSPREFETCH in the "Environment Variables" chapter of *Environment Variables & System Options* for more information on setting this environment variable.

For additional information on prefetching, see READS in the "Synergy DBL Statements" chapter of the *Synergy DBL Language Reference Manual*.

For ISAM files, you can further improve performance by using the /bufstore option in the OPTIONS qualifier on the OPEN statement to buffer records on a STORE when system option #36 is not set. See OPTIONS in the "Synergy DBL Statements" chapter of the *Synergy DBL Language Reference Manual*.

Using Client/Server Encryption

The xfServer encryption feature enables you to encrypt the transfer of sensitive data across a network. xfServer interfaces with a third-party library, OpenSSL, to provide SSL support for secure data transport between client and server.

To use encryption, the server and all clients must be version 9.3 or higher. To be able to specify the protocol version (**-scl** option), the server and all clients must be version 10.3.1b or higher.

You have the option of using master or slave encryption. When *master* encryption is enabled, *all* data packets to and from the server are encrypted. When *slave* encryption is enabled, data packets on specific channels are encrypted. You can indicate what data should be encrypted with an option on the OPEN statement or by flagging a particular file as requiring encryption. See "Specifying the data to encrypt for slave encryption" on page 3-9.

TIP

Using encryption can affect performance because data must be encrypted and decrypted on both sides of the xfServer connection. The cipher negotiation between client and server, even though it happens only on the first OPEN statement, also takes time.

Although master encryption is inherently more secure, slave encryption affords better performance. As an alternative to slave encryption, you may want to consider having two xfServer services—one that uses master encryption for secure communication and a second one for non-secure communication.

Whether using master or slave, you can improve performance by using Select statements to read the file on the server and select the records to send to the client; then, only the selected records need to be encrypted and decrypted, rather than the whole file. (See Synergex.SynergyDE.Select.Select in the "System-Supplied Classes" chapter of the *Synergy DBL Language Reference Manual* for more information on the Select class.) Another way to improve performance is to use xfServerPlus to do processing on the server, and then send only the necessary data to the client. (See the *xfNetLink & xfServerPlus User's Guide*.)

To implement encryption, you must start **rsynd** with the **-encrypt** option (/ENCRYPT on OpenVMS) or by selecting the Enable encryption option in the Synergy Configuration Program on Windows. You also specify a certificate file, the cipher level, and the security compliance level (i.e., protocol). The three cipher options (high, medium, low) map to specific cipher suites and protocols, which vary by OpenSSL version. See "Setting up the server and client machines for encryption" on page 3-10.

If you see the error "Cannot load random state", it means there is not enough random data on the system to seed cryptographic algorithms. To correct this, you must define the SYNSSL_RAND environment variable on either the client or the server (depending on where the error occurred) to point to a file that can be used to gather random data. (See SYNSSL_RAND in the "Environment Variables" chapter of *Environment Variables & System Options* for more information.)

Understanding cipher suites and protocols

When enabling encryption for xfServer, you specify a cipher level, which determines which cipher suites will be used, as well as a security compliance level (**-scl** option), which defines the protocols to use.

The cipher levels (low, medium, high) map to specific cipher suites, which are grouped into security levels by OpenSSL. These suites and their groupings vary by operating system and OpenSSL version, and they often change as new, more secure ciphers are developed and older ones are found to be vulnerable and are retired. This is why **rsynd** requires only the level of cipher suite desired rather than a specific suite. (You can see the specific suite and protocol being used; see "Verifying that encryption is requested" on page 3-11.)

Each cipher suite supports one or more protocols, such as SSLv3 or TLS 1.0. Support for TLS 1.2, which is, as of this writing, the highest protocol available, was added to xfServer in 10.3.1b. xfServer also supports TLS 1.1 and 1.0. As of 10.3.1b, SSLv3 is no longer supported.

Bear in mind that because protocol and cipher support varies by operating system, older systems may not support the newer protocols. For example, Windows Vista, Windows Server 2008, and HP-UX (PA-RISC) do not support any protocols above TLS 1.0. If you have older clients connecting to xfServer, you will likely need to use level 1 (**-scl=1**), which supports TLS 1.0, 1.1, and 1.2, and should work with all clients dating back to version 9.3 (which is when encryption was introduced). Specifying security compliance level 2 (**-scl=2**) restricts support to TLS 1.1 and 1.2, which may be required for compliance with certain industry security standards. Setting the security compliance level to zero (**-scl=0**) causes **rsynd** to always use the default security compliance level, whatever it may be. This means the protocols available may change when xfServer is upgraded. (As new threats and vulnerabilities are found, Synergex may need to update the default protocols to maintain a high level of security.) See the **-scl** option in "The rsynd Program" on page 3-44 for the current default.

You can also specify a security compliance level on the xfServer client with the **/scl=*level*** runtime option to the OPEN statement. The scl value on the client should be the same as specified on the server because the client and server must have a cipher suite and protocol in common in order to establish secure communication. By setting an scl value on the client, you can ensure that the server that the client connects to is running at the desired security level. (See OPTIONS in the "Synergy DBL Statements" chapter of the *Synergy DBL Language Reference Manual* for more information about using the **/scl** option with the OPEN statement.)

If the client is unable to negotiate a protocol with the server, the server will close the socket and log an error to the event log (Windows) or rsynd log (UNIX and OpenVMS). On the client, the Synergy Runtime will generate the error "Client server error: host: *name*" ($ERR_CLNTERR). This error can be generated for any number of reasons; the informational text will tell you the problem. When the problem is an inability to negotiate a protocol, you'll see "Error requesting secure connection: socket error: an EOF was observed that violates the protocol".

Specifying the data to encrypt for slave encryption

When xfServer has been started with slave encryption, you can control what data is encrypted in the following ways:

- Use the **/encrypt** runtime option for the OPEN statement. The **/encrypt** option causes the client to request data packet encryption from the server. If encryption is not enabled on the server, a “Client server error, host:*xxx*” ($ERR_CLNTERR) occurs. (If master encryption is enabled on the server, the option is ignored, since all data is encrypted anyway.)

  There is also an **/scl** runtime option for the OPEN statement that can be used to specify the desired protocol.

  See OPTIONS in the “Synergy DBL Statements” chapter of the *Synergy DBL Language Reference Manual* for more information about **/encrypt** and **/scl**.

- (Windows and UNIX only) Set the network encryption flag on specific files. You can do this with the NETWORK_ENCRYPT option to the ISAMC routine or with the **-qfile=network_encrypt** option to the patch command (**-p**) in the **isutl** utility.

  This is the recommended method because it ensures that *any* client that accesses that file *must* use encryption. If you have a file with sensitive data, it is more efficient, as well as more secure, to set the network encryption flag, than to add the **/encrypt** option to every OPEN statement that opens the file. You can still specify the **/scl** option if needed on the OPEN statement; if not specified, the default is used.

  If a file with the network encryption flag set is referenced with an OPEN statement and encryption has not been enabled on the server, a “File requires network encryption” error ($ERR_NETCRYPT) occurs. 0

  The network encryption flag applies to *all* remote access from a Synergy application. This means that attempting to access the file via a network path specification (e.g., a mapped drive on Windows) will generate the error $ERR_NETCRYPT. There is no prohibition to accessing the file locally.

  For more information about setting the network encryption flag, see ISAMC in the “System-Supplied Subroutines and Functions” chapter of the *Synergy DBL Language Reference Manual* and isutl in the “Synergy DBMS” chapter of *Synergy Tools*.

Setting up the server and client machines for encryption

Follow these steps to set up your xfServer system to use encryption. For details on which version of OpenSSL is required for your operating system, see "OpenSSL Requirements" on page 1-11. For additional information see Synergex KnowledgeBase article 100001979.

1. Install OpenSSL on your server machine.
   - For Windows, go to www.openssl.org/community/binaries.html and follow the link to download the version for your platform. See Synergex KnowledgeBase article 100001979 for details on file placement.
   - For UNIX, use the OpenSSL libraries installed with your operating system or download the OpenSSL security kit specific to your platform and version from your operating system vendor.
   - For OpenVMS, use the operating system-supplied libraries, HP SSL, available from h71000.www7.hp.com/openvms/products/ssl/ssl_download.html.
2. Ensure that the OpenSSL shared libraries are in the correct location on the server or have been added to the correct path. The library path must be set *before* registering **rsynd** on Windows or starting **rsynd** on UNIX and OpenVMS.
   - On Windows, the libraries must be located on the xfServer machine in the dbl\bin directory that corresponds to the bitness of your machine (as explained in KB 100001979). If the libraries cannot be found, the operating system generates the error "Encryption is required but not available. A service specific error occurred: 14".
   - On UNIX, the OpenSSL libraries are installed in a standard location determined by the operating system. You don't need to move them or set a path. If the libraries are not installed, the operating system generates the error "synssllib.so not available".
   - On OpenVMS, during installation, SYNSSLLIB is set in SYS$MANAGER:SYNERGY_STARTUP.COM to the full path and filename of **synssllib.exe**, the SSL runtime support file. If **synssllib.exe** cannot be found, **rsynd** generates the error "SYNSSLLIB not set".
3. Create a certificate file (**.pem** file). If you name this file **rsynd.pem** and put it in DBLDIR, it will be used by default when you start **rsynd** with encryption enabled. However, you may name the file anything you like and put it elsewhere if desired, and then specify it with the **-cert** option (/CERTIFICATE on OpenVMS).

   On Windows and OpenVMS, the certificate file *cannot* include a passphrase. On UNIX, a passphrase is permitted.

> See "Requesting a certificate" in the "Synergy HTTP Document Transport API" chapter of the *Synergy DBL Language Reference Manual* for instructions on creating a certificate request file and sending it to a public certificate authority (CA). The CA will then send back a certificate file. You may decide that you do not need a public certificate since xfServer is a proprietary format used on a local server, in which case the generation of a local CA with self-signed certificates may suffice. See "Testing your HTTPS setup locally" in the above-mentioned chapter for steps on how to create the file.

4. Install and configure OpenSSL on the client machines.
   - On Windows clients, install OpenSSL in the dbl\bin directory.
   - On UNIX clients, the library path is used to find the OpenSSL libraries. If you used the **setsde** script to set up your Synergy environment, the path will be correctly set. You can run the **dltest** utility (installed in the Synergex\SynergyDE\connect directory) to find your shared library path and determine if Synergy can find the necessary DLLs.
5. Start **rsynd** with the **-encrypt** option (/ENCRYPT on OpenVMS) and specify master encryption if desired. (The default is slave.) Specify a non-default certificate filename, cipher level, and security compliance level if desired. See "The rsynd Program" on page 3-44 for detailed syntax and examples.

   Or, on Windows, start **rsynd** from the Synergy Configuration Program and select the Enable encryption option, and then specify the cipher level, certificate file, and security compliance level. See "Modifying the SynSrv xfServer service" on page 3-12.

> If you are running xfServer in non-secure mode (**-n** or /NOSECURE) with encryption enabled, you *must* specify a default user account (**-u** or /DEFAULT_USER).

Verifying that encryption is requested

To verify that a channel is encrypted, use the SLE (Socket Level Encryption) option to the GETFA routine; it returns 1 or 0.

To see the specific cipher and protocol being used, use the SLC (Socket Level Cipher) option to the GETFA routine; it returns a string identifying the cipher and protocol. For example,

```
AES256-GCM-SHA384     TLSv1.2  Kx=RSA      Au=RSA   Enc=AESGCM(256)  Mac=AEAD
```

If the *field* argument is not large enough to hold the entire string (which could be up to 128 bytes), it will be truncated. If the channel is not encrypted, a blank string is returned. For more information on the SLE and SLC options, see GETFA in the "System-Supplied Subroutines and Functions" chapter of the *Synergy DBL Language Reference Manual*.

If you are using file level encryption, you can verify that a file is encrypted with the NETCRYPT request keyword to the %ISINFO function. See %ISINFO in the "System-Supplied Subroutines and Functions" chapter of the *Synergy DBL Language Reference Manual*.

Using xfServer on Windows

During installation, xfServer is installed with default settings for port, logging, security, and connection recovery. The installation program also registers xfServer in the Windows registry using the default service name (**SynSrv**) and port (2330), and then starts it.

You can modify the default settings for **SynSrv** or add additional xfServer services using the Synergy Configuration Program.

When you're running xfServer, you can use the Monitor utility (**synxfmon.exe**) to find out which files are open, who opened them, whether those files are locked, as well as whether any client contexts are being held. See "The Monitor Utility for Windows" in the "General Utilities" chapter of *Synergy Tools*.

Modifying the SynSrv xfServer service

Before modifying any xfServer service, verify that no users are currently connected. The service must be stopped and restarted for changes to take effect, so any users would be disconnected. In addition, if connection recovery is enabled and there are client contexts being held, they will be deleted.

You must be logged on using an account that has administrator privileges to modify settings and to register and start an xfServer service.

Do not attempt to issue **rsynd** commands from the command line while the Synergy Configuration Program is running.

1. Start the Synergy Configuration Program (from the Windows Control Panel, select Synergy Control Panel > Synergy Configuration Program) and go to the xfServer/xfServerPlus tab.
2. Select the service **SynSrv** in the list of services and click the Modify Service button.
3. Modify the fields in the xfServer Information dialog box. You cannot change the service name. If you do not want to use the default service name, you must remove this service and add a new one with the desired name.

   Port number. Enter the port you want xfServer to run on. Valid ports are in the range 1024 through 65535. The default is 2330.

   If you use a non-default port for the server, you must use SCSPORT to specify it on the clients. See SCSPORT in the "Environment Variables" chapter of *Environment Variables & System Options* for more information.

   Display name. Enter a display name for this service. This is the name that displays in the Windows Services console. If you leave this field blank, it defaults to "Synergy/DE xfServer ####", where #### is the port number.

Secure/Restricted/Non-secure. Indicate if you want to run xfServer in secure mode, restricted mode, or non-secure mode. The default is secure. See "Understanding xfServer security" on page 3-16 for a detailed explanation of security modes.

- **Secure.** Secure mode gives you the option of using RUSER security or Windows authentication security. Windows or UNIX clients with RUSER set to a user name and password will be authenticated using those credentials. Windows clients without RUSER set will be authenticated using Windows authentication.
- **Restricted.** Restricted mode uses Windows authentication exclusively; consequently, only Windows clients are allowed. If RUSER is set on the client, it is ignored.
- **Non-secure.** No user name or password is used from the client. All clients will run under the service account, which, by default, is the SYSTEM account. You can specify a different account for clients to run under by selecting the "Allow anonymous users" option below.

Allow anonymous users on default user account. Select this option if you want to specify a default user account for anonymous access when running in non-secure mode. For more information on this option, see "Non-secure mode" on page 3-18. If you plan to enable encryption, you must specify a default user for non-secure data access.

Username. If you selected "Allow anonymous users", enter the user name for the default user account that you created to use with xfServer. This account can be either a local account or a domain controller account; it cannot be a member of the administrators group. If there is an account with the same user name on both the local machine and on a Windows domain, or on multiple domains, and you want to use a specific domain account, you must explicitly specify the domain name in the format *user_name@domain_name* or *domain_name\user_name*.

Password. If you selected "Allow anonymous users", enter the password for the account that you created to use with xfServer.

Compress data packets. Indicate whether you want to compress data records sent between xfServer and its clients. This option compresses blanks, nulls, zeros, and repeating characters. Compression can significantly improve performance on low speed or busy networks, especially WANs.

Compression can also be set on the client with the SCSCOMPR environment variable. To turn compression off, it must be turned off on both the server and the client. For more information, see SCSCOMPR in the "Environment Variables" chapter of *Environment Variables & System Options*.

Enable logging. Select this option to enable event logging, which logs all user connections. **Rsynd** always logs its version, start/stop information, and errors to the event log, regardless of whether this option is enabled. To view event logs, use the Windows Event Viewer and choose the application event log.

Verbose logging. Select this option to enable more verbose event logging. In addition to user connections, the event log will also show informational messages, which may be helpful when troubleshooting. Due to the number of events that could potentially be logged, you will probably want to turn off verbose logging when you are done troubleshooting.

Enable encryption. Select this option to enable encryption of data between client and server, and then select the type of encryption desired, Slave or Master. See "Using Client/Server Encryption" on page 3-7 for details on this feature.

Certificate file. Specify the certificate file you created using either the full path or a logical. The default filename is **DBLDIR:rsynd.pem**, but you may choose another name and place the file anywhere you like. (Note that the resolved path displays rather than "DBLDIR".)

Cipher type. Select the level of encryption desired. These levels map to specific cipher suite names and protocols, which are determined by the version of OpenSSL being used. If Medium or Low is specified, Security level must be set to "TLS 1.0 + TLS1.1 + TLS 1.2"**.** You can use the SLC option to the GETFA routine to see which cipher suite and protocol are actually used. See GETFA in the "System-Supplied Subroutines and Functions" chapter of the *Synergy DBL Language Reference Manual*.

Security level. Displays the currently selected protocols. To change this setting, click the Change button to display the Security Compliance Level dialog. The default is "Always use current default", which means that *the available protocols may change with the version of Synergy.* If you don't want the available protocols to change when you upgrade xfServer, clear the check box and explicitly select the set of protocols you'd like to be available. Selecting "Level 2: Use TLS1.1 + TLS 1.2" requires a cipher type of High. See "Understanding cipher suites and protocols" on page 3-8 for more information.

Enable connection recovery. Select this option to enable the client to recover the connection and the session context after an unexpected socket disconnect, and then select the type of connection recovery desired, Slave or Master. See "Using connection recovery" on page 3-19 for details on this feature.

Select profile. Select the desired connection recovery profile, Default, Mobile, or Mobile2 to indicate the time, in seconds, for the four connection recovery parameters. Alternatively, select Custom and define your own set of time values.

4. If you want to modify environment variables for xfServer, click the Environment Settings button and see "Defining environment variables for xfServer services" on page 3-25 for instructions.

TIP

If you create files without specifying a path (i.e., you rely on a default path), you may want to set the RSFILPATH environment variable to indicate where you want the files to go. You can use the Synergy Configuration Program to set RSFILPATH. For more information about this environment variable, see RSFILPATH in the "Environment Variables" chapter of *Environment Variables & System Options*.

5. Click OK in the xfServer Information dialog box.

6. Click Apply in the Synergy Configuration Program. If the service is currently running, the Synergy Configuration Program stops it and then restarts it with the new settings. See the “xfServer Startup and Shutdown Codes” table on page 3-27 for status and error codes that may occur.

Adding a new xfServer service

You can run multiple xfServer services; each service must have a different port, service name, and display name. You must have administrator privileges to register and start an xfServer service.

1. Start the Synergy Configuration Program and go to the xfServer/xfServerPlus tab.
2. Click the Add xfServer Service button.
3. In the Service name field, enter a name for this service. This name will display on the list of services on the xfServer/xfServerPlus tab and in the registry.
4. Values in the remaining fields default from the <Default> entry. Change them as necessary. See step 3 on page 3-12 for detailed field information and see “Using the <Default> entry” on page 3-15 for information on setting default values for new services.
5. If you want to modify environment variables for xfServer, click the Environment Settings button and see “Defining environment variables for xfServer services” on page 3-25 for instructions.
6. Click OK. The new service displays in the list of services.
7. Click Apply to register the new service.
8. If desired, start the service now by clicking the Start Service button. You can also start the service later; see “Starting xfServer” on page 3-27.

Using the <Default> entry

The <Default> entry, which displays in the Services list on the xfServer/xfServerPlus tab of the Synergy Configuration Program, can be used to set default values for some xfServer and xfServerPlus settings and for environment variables used by xfServer and xfServerPlus. To modify the settings for the <Default> entry, select <Default> in the list of services and click the Modify button.

- The following settings are applied to all newly created xfServer services. Existing services are not affected.
  - The type of data access—Secure, Restricted, or Non-secure (but not the “Allow anonymous users” checkbox or the username and password for the default user account)
  - The Compress data packets option
  - The Enable logging and Verbose logging options
  - All Connection recovery settings

- The following settings are applied to all newly created xfServer and xfServerPlus services. Existing services are not affected.
  - The Enable encryption option, the Certificate filename, the Cipher type, and the Security level.
- Environment variables are applied to all existing xfServer and xfServerPlus services, as well as to any new services that you create. See "Defining environment variables for xfServer services" on page 3-25 for more information on environment variables.

Understanding xfServer security

On Windows, xfServer can be run in secure mode, restricted mode, or non-secure mode.

What are RUSER security and Windows authentication security?

RUSER security secures the connection between client and server using user name and password credentials supplied by the client. The credentials are passed to xfServer for authentication. The user name will be used to generate the persona used during xfServer access. This is referred to as "RUSER security" because the credentials are defined on the client using the RUSER environment variable or registry setting.

⚠ Even though they are encoded, RUSER credentials should be kept confidential because they can be used with any xfServer client.

Windows authentication security uses the Windows operating system to authenticate Windows clients. No credentials are passed from the client to xfServer. Rather, the local security authority is used to establish a security context between client and server. We recommend that you use Windows authentication security whenever possible.

Secure mode

Overview

Secure mode enables you to use both RUSER security and Windows authentication security.

On either a Windows or UNIX client, if the RUSER environment variable (or registry setting) is set to a user name and password, RUSER security will be used. The client sends the user name and encoded password to the server, where it is checked against the user's name and password on the server. If authentication fails, an error will be generated and access to xfServer will be denied.

On a Windows client, if RUSER is not set, Windows authentication will be used. Windows authentication uses the same security as is used when a user logs into a domain account and has access to the resources in that domain. A Windows client logged into a domain can have access to any xfServer running within that domain. Access to individual files and folders is controlled by the user's account on the domain, as set up by the Windows system administrator. No credentials are passed to the server. If Windows authentication fails, access to xfServer will be denied.

On a UNIX client, if RUSER is not set, an error will be generated and access to xfServer will be denied.

Benefits

Secure mode is of particular benefit when you have both Windows and UNIX clients connecting to a Windows server. The Windows clients can use Windows authentication, requiring less setup, while the UNIX clients can use RUSER security.

Setting up

To use RUSER security, you must run the **setruser** utility on each client. On a Windows client, **setruser** will generate the encoded password for RUSER and set RUSER in the registry. (RUSER can also be set in the environment on Windows; the environment setting takes precedence over a registry setting.) On a UNIX client, **setruser** will generate the encoded password for RUSER, which you can then use to set the RUSER environment variable. In addition, there must be a matching user name and password on the server machine or on a Windows domain where the server machine is a member. (For more information, see “The setruser Utility” on page 3-66 and RUSER in the “Environment Variables” chapter of *Environment Variables & System Options*.)

To use Windows authentication, the client user must have an account on a Windows domain, and the xfServer machine belongs to the same domain or trusted domain. Ideally, RUSER should not be set. However, if there is a global RUSER setting on the client machine, you can ensure that Windows authentication is used instead of RUSER security by running **setruser** and specifying “SSPI” for the user name (don’t specify a password; just press RETURN when prompted for it).

Running the server

In the Synergy Configuration Program, select “Secure” for data access. This is the default. (See “Modifying the SynSrv xfServer service” on page 3-12 for details.)

If you are starting xfServer from the command line, you can specify the **-s** option when you register the service, or you can just not specify a security option at all, as **-s** is the default. (See “The rsynd Program” on page 3-44 for details on command-line options.)

Restricted mode

Overview

Restricted mode uses Windows authentication exclusively. Consequently, the server and all the clients must be Windows machines. Once a user has successfully logged into the Windows domain, that user can be granted access to any xfServer within the domain. Access to individual files and folders is controlled by the user’s account on the domain, as set up by the Windows system administrator. If RUSER is set, it is ignored.

Benefits

One of the advantages to restricted mode, in addition to the ease of implementation, is that you can guarantee that no user credentials (user name and password) are sent over the network.

Setting up

There is no special set-up required on the clients; the clients and the xfServer machine must all be on the same domain.

Running the server

In the Synergy Configuration Program, select “Restricted” for data access. (See “Modifying the SynSrv xfServer service” on page 3-12 for details.)

If you are starting xfServer from the command line, specify the **-sspi** option when you register the service. (See “The rsynd Program” on page 3-44 for details on command-line options.)

Non-secure mode

Overview

Non-secure mode does not require an RUSER setting on the client. If RUSER is set, it is ignored. All clients take on the persona of the service, which is usually the SYSTEM account. You can specify a different account by running in non-secure mode with a default user account, which can be set up to restrict users’ access to the server. When you specify a default user account, all clients take on the persona of the default user. If you are going to use non-secure mode, we recommend that you use a default user account.

Benefits

Using non-secure mode with the default user account is simple to set up and restricts users’ access on the server. It can be used with both Windows and UNIX clients.

Setting up

No special setup is required on the client side or the server side to run in non-secure mode without a default user account.

To use the default user account option, you will need to create an account. We recommend that you set up an account with limited privileges specifically for use with xfServer. This account can be on the xfServer machine (a local account), or it can be a domain controller account. The account cannot be a member of the administrators group.

> If you are running xfServer in non-secure mode with encryption enabled, you *must* specify a default user account.

The account that you create for use with xfServer must allow a log-in on the machine that xfServer is running on. If you cannot log on to the xfServer machine using the account, verify that the account has the “log on locally” user right set. Sometimes, if the account is a domain controller account and xfServer is running on the same domain controller, this user right will not be set.

Running the server

In the Synergy Configuration Program, select “Non-secure” for data access. To use a default user account, select the “Allow anonymous users on default user account” check box, and then enter the user name and password for the account you created for use with xfServer. (See “Modifying the SynSrv xfServer service” on page 3-12 for details.)

If you are starting xfServer from the command line, specify the **-n** option when you register the service. To use a default user account, also specify the **-u** option, followed by the username/password of the xfServer account, when you register the service. (See “The rsynd Program” on page 3-44 for details on command-line options.)

Using connection recovery

The connection recovery feature enables an xfServer client application to seamlessly reconnect to the server and recover its session context after an unexpected loss of connectivity.

To use connection recovery, both the server and clients must be version 10.3 or higher. This feature is supported only for Windows clients connecting to a Windows server. Older Windows clients and UNIX clients can connect to a Windows server that has connection recovery enabled, but they cannot take advantage of it.

This feature is intended primarily for wireless networks, which may include laptop computers or mobile devices such as tablets, rather than for traditional, wired networks where the clients are desktop computers and loss of connectivity is rare. For example, in a large warehouse with overlapping Wi-Fi networks, a device running an application with a connection to xfServer would lose that connection as the user moved about the warehouse, going in and out of range of the various wireless network access points. And when the connection is lost, all contexts for the application are also lost, including which record in which file was being read, as well as records locked for updating. When this happens, the user gets an error and has to restart the application, determine where he was and what he was doing, and perhaps re-enter data.

With the connection recovery feature, however, when the connection is unexpectedly dropped, the client context will be saved on the server and the client application will attempt to reconnect. If it is able to reconnect within a certain time limit (which you can specify), the client context can be recovered. In the warehouse scenario, the end user would be unlikely to even notice what had happened. To handle the possibility of a longer disconnect period, you can add code to your application to ask the user whether to continue trying to reconnect.

How connection recovery works

Connection recovery can be run in master mode or slave mode. When *master* mode is enabled, the feature is enabled for all clients, but can be disabled on individual clients. When *slave* mode is enabled, activation of the feature is controlled by setting the SCSKEEPCONNECT environment variable on the client. (For devices such as tablets, where environment variables are not supported, connection recovery is implicitly activated on the client.) In both master and slave mode, you can control the length of time that the client continues to try to connect, how long client context is maintained, and how long (or if) records and files are kept locked. By default, connection recovery is enabled in slave mode with default settings for these time values.

To completely disable connection recovery, on the xfServer machine clear the Enable connection recovery check box in the Synergy Configuration Program or start **rsynd** from the command line with the **-keepconnect=off** option.

A disconnected socket may be detected on the server when it pings to see if an idle client is still present, or it may be detected by the client when it makes a request to the server. Once the dropped socket is detected, the client context is saved on the server machine and held for the length of time defined by the KeepContext value. If KeepLocks is non-zero, record and file locks will be maintained for the KeepLocks value. If KeepLocks is zero, record and file locks are released immediately when the disconnect is detected.

If the client detects the dropped connection, it will continue trying to reconnect to the server, making a request every 5 seconds for the length of time defined by the RetryTime value. Once that time has expired, a “Server connection retry failure” error ($ERR_SRVCONTRY) will be generated. However, if the KeepContext time has not expired, the context will continue to be maintained and can be recovered by the client if it reconnects.

It is important to understand that the KeepContext, KeepLocks, and RetryTime timers all start *after* the socket disconnect has been detected. How long this takes varies depending on your network and the Keepalive setting and whether the disconnect is detected by the server or the client. On the server side, a disconnect cannot be detected until the TCP keepalive timer fires. For this reason, you may want to set KeepAlive to a fairly short period so that dropped sockets are detected quickly. By default, KeepAlive is set to 150 seconds (two-and-a-half minutes).

Implementing connection recovery in your application

You can use this feature straight out of the box with the default settings without making any changes to your application. Be aware, however, that $ERR_SRVCONTRY, which occurs when the RetryTime expires, is a new error that your application may not be trapping for. To take full advantage of connection recovery, you may want to make some changes to your application.

Users actively using your application are unlikely to patiently wait a minute or two for a connection unless they know that the application is attempting to reconnect. One way to handle this is to trap for $ERR_SRVCONTRY and reissue the request. A simpler way to keep the user informed is to use the IOHooks reconnect_hook method. This method is called when the RetryTime value expires, prior to the $ERR_SRVCONTRY error being generated. Returning TRUE from reconnect_hook re-issues the failed request and resets RetryTime. Returning FALSE terminates the retry request and the $ERR_SRVCONTRY error is generated. For example, you could write a method that displays a message dialog that indicates the application is attempting to connect and offers the user the option to continue or terminate. For more information on using the IOHooks class, see Synergex.SynergyDE.IOExtensions.IOHooks in the “System-Supplied Classes” chapter of the *Synergy DBL Language Reference Manual.*

Understanding the connection recovery options

You can set these values on the server from the Synergy Configuration Program or the command line. On the client, you can use the environment variables SCSKEEPCONTEXT, SCSKEEPALIVE, and SCSKEEPLOCKS to override the values on the server for that client. There is no client-side environment variable to override RetryTime, but you can do so by setting one of the profiles (see "Understanding the predefined profiles" on page 3-23).

KeepContext

Defines the length of time, in seconds, that xfServer will maintain the context for a client connection after an unexpected socket disconnect has been detected. Note that due to the keepalive timer, the disconnect may not be detected immediately, especially if the client is idle. If xfServer receives a reconnection request from the client within the KeepContext time, it will resume the suspended connection with the last known context, including open files, file position, and (depending on the KeepLocks setting) file and record locks. If the client does not reconnect within the specified time, the context is deleted and all file and record locks are released. The xfServer license is held with the context and not released until the KeepContext time expires.

TIP You can use the **synxfmon** utility to view suspended connections and how long they have been suspended. You can also terminate suspended connections with **synxfmon**. See "The Monitor Utility for Windows" in the "General Utilities" chapter of the *Synergy Tools* manual for more information on **synxfmon**.

KeepAlive

Defines the TCP keepalive timer setting when connection recovery is enabled. The keepalive timer fires when there has been no communication from the client for the length of time specified by the keepalive timer. For connection recovery, we recommend a fairly short keepalive time, so that disconnected sockets are detected quickly and the context saved. But this efficiency comes at the cost of more frequent TCP/IP pings, which may be undesirable in a busy network. For more information about the keepalive timer, see "Rsynd: shutting down and timing out" on page 3-71.

KeepLocks

Defines the length of time that records and files remain locked when a client context is saved. By default, records and files that are locked when the client context is saved remain locked for the length of time specified with the KeepLocks value. When KeepLocks expires, record locks are released. Files opened with exclusive locks (SHARE:Q_EXCL_RO or SHARE:Q_EXCL_RW) are released and closed. Files opened for output or append, however, remain locked until KeepContext expires or the client regains control of the context.

Note that files must be closed on the server to release the locks. Upon reconnecting, the client will receive a "Remote file has been closed" error ($ERR_SRVFILCL). Your application will need to handle re-opening the file.

TIP We recommend using temporary filenames (or some other type of unique filenames) for output and append files such as log files. Otherwise, a restarted application might encounter a locked file.

All locks are released when KeepContext expires. Consequently, the KeepLocks time cannot be longer than the KeepContext time. If KeepLocks is shorter than KeepContext, a client that reconnects after locks have been released (but before the context expires) will need to relock records prior to updating; otherwise, a "No current record" error ($ERR_NOCURR) will be generated. If you do not want files and records to remain locked when a context is saved, set KeepLocks to 0 (zero). Even if KeepLocks is 0, files opened for output or append with exclusive locks will remain locked until the KeepContext time expires.

TIP Take care when choosing a KeepLocks value. If your application uses optimistic locking (which we recommend), KeepLocks can be set to a very low value—even zero. But if your application holds on to record locks, you'll probably want a larger value. In most cases, a relatively short time—no more than 5 minutes—should suffice. A longer time runs the risk that the user may give up waiting for the socket to reconnect, exit and restart the application, and then discover that records cannot be accessed because they are still locked.

RetryTime (client retry)

Defines the length of time that the client will continue to attempt to reconnect to a server thread when it detects that it has been disconnected. When a disconnected socket is detected by the client, it automatically attempts to reconnect, and will continue to make connection requests every 5 seconds for the length of RetryTime before generating a "Server connection retry failure" error ($ERR_SRVCONTRY).

It is not necessary to make RetryTime as long as the KeepContext time because xfServer is continuously watching the socket, whereas the client application will detect a bad socket only when it attempts I/O. Chances are that the server will already be waiting for the client's reconnect request, so it's the KeepContext time that you need to set to indicate how long you want the context to be maintained.

Understanding the predefined profiles

There are three predefined profiles—Default, Mobile, and Mobile2—which simply offer a convenient way to set the four time values used by connection recovery. You can use these profiles on the server by selecting one in the Synergy Configuration Program (profiles are not available when starting **rsynd** from the command line) or on the client by specifying one with the SCSPROFILE environment variable. The profiles are defined as follows (times are in seconds):

Connection Recovery Profile Definitions

Setting	Default profile	Mobile profile	Mobile2 profile
KeepContext	300	7200	7200
KeepAlive	150	60	60
KeepLocks	300	300	300
RetryTime	60	60	20

The Default profile represents the built-in connection recovery defaults.

The Mobile profile is suitable for use with an application that has no special handling for socket failures. It provides a reasonable time to re-establish a failed connection, so it's a good choice if you are using connection recovery out of the box without any modifications to your application.

The Mobile2 profile is suitable for use with an application with built-in awareness of socket failures, which enables the application to control what action to take after a failure. This awareness may be in the form of an IOHooks reconnect_hook method or it could be code that reissues the request upon trapping a "Server connection retry failure" ($ERR_SRVCONTRY) error.

Using connection recovery in slave mode

In slave mode, the connection recovery feature is activated only when the client requests it. The advantage of slave mode is that it lets you determine which clients on your network need the feature. For example, your network may include both wired desktop systems (which rarely, if ever, experience a socket disconnect) and laptops using Wi-Fi (which may experience socket disconnects as they are moved around the office).

1. On the xfServer machine, in the Synergy Configuration Program, select the Enable connection recovery check box and the Slave radio button.
2. Choose a profile or select Custom and specify your own time values.
3. Set the SCSKEEPCONNECT environment variable to ON on clients that you want to use connection recovery. Leave SCSKEEPCONNECT unset or set it to OFF on clients that you do not want to use the feature.

4. By default, clients use the connection recovery settings on the server. To override these settings, set the SCSKEEPCONTEXT, SCSKEEPALIVE, and SCSKEEPLOCKS environment variables on the client. Alternatively, you can set SCSPROFILE on the client to specify a predefined set of values.

> TIP When connection recovery is enabled in slave mode, setting SCSKEEPCONTEXT, SCSKEEPALIVE, SCSKEEPLOCKS, or SCSPROFILE on the client will enable the feature for that client, as long as SCSKEEPCONNECT is not set to OFF.

Using connection recovery in master mode

When connection recovery is enabled in master mode, the feature is activated for all clients, unless individual clients request that it be disabled.

1. On the xfServer machine, in the Synergy Configuration Program, select the Enable connection recovery check box and the Master radio button.
2. Choose a profile or select Custom and specify your own time values.
3. By default, clients use the connection recovery settings on the server. To override these settings, set the SCSKEEPCONTEXT, SCSKEEPALIVE, and SCSKEEPLOCKS environment variables on the client, or use the SCSPROFILE environment variable to specify a predefined set of values.
4. To disable connection recovery for individual clients, set the SCSKEEPCONNECT environment variable to OFF on the client.

See also

- SCSKEEPCONNECT in the “Environment Variables” chapter of *Environment Variables & System Options*.
- SCSKEEPCONTEXT in the “Environment Variables” chapter of *Environment Variables & System Options*.
- SCSKEEPALIVE in the “Environment Variables” chapter of *Environment Variables & System Options*.
- SCSKEEPLOCKS in the “Environment Variables” chapter of *Environment Variables & System Options*.
- SCSPROFILE in the “Environment Variables” chapter of *Environment Variables & System Options.*

Defining environment variables for xfServer services

See "Why use environment variables with xfServer?" on page 3-4 for a general discussion of the benefits of using environment variables with xfServer. If your client application includes environment variables in the format "LOGICAL:@server_name", you must define those environment variables on the server, where xfServer can find them.

The recommended method for defining environment variables for xfServer is to use the Synergy Configuration Program, which writes the settings to the Windows registry. (xfServer cannot read environment variables set in the environment; they must be set in the registry.) Using the Synergy Configuration Program, you can define environment variables for all xfServer services or for a specific instance of xfServer. Environment variables can also be defined for a specific user, but you must do so by manually editing the registry; see "Defining environment variables for a specific user" on page 3-26.

Before adding or changing environment variables, you should verify that no users are currently connected. The service must be stopped and restarted for environment variable changes to take effect, so any users would be disconnected and any saved client contexts lost.

Defining environment variables for all services

Environment variables set in this manner will apply to all instances of xfServer and xfServerPlus, both existing and newly created.

> If an environment variable is already set for a specific service (see "Defining environment variables for a specific service" on page 3-26), and you set that same environment variable for "all services", the service-specific setting will *not* be overridden. If your goal is to use the "all services" setting, you must delete the service-specific setting.

1. Start the Synergy Configuration Program and go to the xfServer/xfServerPlus tab.
2. Select <Default> from the list of services, click the Modify Service button, and then click the Environment Settings button. Only the top portion of the Environment Settings dialog box is enabled.
   - To **add** a new setting, click the Add button. Type the variable name and value in the Add Environment Setting dialog box and click OK.
   - To **modify** an existing setting, select it in the list of variables and click the Modify button. Make changes in the Modify Environment Setting dialog box and click OK.
   - To **remove** an environment variable, select it in the list of variables and click the Remove button.
3. When you are through, click OK in the Environment Settings dialog box.
4. Click OK in the xfServer Information dialog box, and click Apply in the Synergy Configuration Program. If any services are currently running, the Synergy Configuration Program will prompt you before stopping and then restarting them with the new settings.

Defining environment variables for a specific service

Environment variables set in this manner will apply only to a specific instance of xfServer and will override settings made for all services.

1. Start the Synergy Configuration Program and go to the xfServer/xfServerPlus tab.
2. Select the desired service, click the Modify Service button, and then click the Environment Settings button. Use the lower portion of the Environment Settings dialog (labeled “Settings for *service name*”) to set environment variables for the selected service.
    - To **add** a new setting, click the Add button. Type the variable name and value in the Add Environment Setting dialog box and click OK.
    - To **modify** an existing setting, select it in the list of variables and click the Modify button. Make changes in the Modify Environment Setting dialog box and click OK.
    - To **remove** an environment variable, select it in the list or variables and click the Remove button.
3. When you are through, click OK in the Environment Settings dialog box.
4. Click OK in the xfServer Information dialog box, and click Apply in the Synergy Configuration Program. If the service is currently running, the Synergy Configuration Program will prompt you before stopping and then restarting it with the new settings.

Defining environment variables for a specific user

Environment variables defined for a specific user will override environment variables set for all xfServer services and those set for a specific xfServer service. The user must have an account on the server machine; it cannot be a domain account. Note that environment variables set at the user level are read when a connection is made. This is in contrast to environment variables set at the server level, which are read when xfServer starts.

You cannot define environment variables at this level from the Synergy Configuration Program; they must be defined manually in the registry. We recommend backing up (exporting) the affected registry branch before making any changes.

1. Log onto the server machine as the user for whom you wish to set the environment variable.
2. Run **regedit**.
3. Navigate to **HKEY_CURRENT_USER\SOFTWARE\Synergex\Synergy xfServer\Synrc**. This registry key is created the first time the user accesses xfServer. If it is not present, you will need to create the key manually.

4. Add a new environment variable.
   - Select Edit > Add Value to display the Add Value dialog box.
   - In the Value Name field, enter the name of the environment variable you want to define.
   - Select REG_SZ for the data type, and click OK.
   - In the String Editor dialog box, enter the path for the environment variable. Click OK.
5. Navigate to **HKEY_LOCAL_MACHINE\ SOFTWARE\Synergex\Synergy xfServer\***service_name***\Default** and set ENABLEUSERHIVE to 1 in that location. This registry setting is required for user-specific environment variable settings.

Starting xfServer

The default service **SynSrv** is started automatically when you install. However, there may be times when you need to start it manually. You must have administrator privileges to start xfServer.

To start xfServer manually, do any of the following:

- Start the Synergy Configuration Program and go to the xfServer/xfServerPlus tab. Select the service from the list and click the Start Service button.
- Go to Administrative Tools > Services. Select the service (the default display name is Synergy/DE xfServer), and select Action > Start.
- At a command prompt enter

  ```
  net start servicename
  ```

  where *servicename* is the name of the xfServer service to be started. The default service name is **SynSrv**.

The following codes may occur when starting or stopping xfServer. Those marked with * are exit status codes only; they do not display, but you can use them to check success or failure conditions programmatically. The others are service-specific codes that can be used programmatically, but may also display in a message window or on the command line under certain conditions.

xfServer Startup and Shutdown Codes

Code	What it means
2	You must be logged on using an account that has administrator privileges to register or unregister the service.
3	The specified account does not exist on the machine.
5	User not found on local machine.

xfServer Startup and Shutdown Codes	
Code	**What it means**
6	The username specified for the default user account for xfServer or for running xfServerPlus sessions belongs to the administrators group. This is not permitted. Check the Windows Event Viewer for more information.
7	No username/password specified. Check the Windows Event Viewer for more information.
8*	Service has been registered.
9*	Service has been stopped.
10*	Service has been un-registered.
12*	Service registration error.
13	Invalid option specified. If SSPI security is enabled (restricted mode), this error may indicate the machine is no longer a part of the domain. Check the Windows Event Viewer for more information.
14	Encryption is required but not available. Check the Windows Event Viewer for more information.
15	Port is in use. It might be the case that you are attempting to start a service that is already running. Check the Windows Event Viewer for more information.

Stopping xfServer

Before stopping or removing (unregistering) a service, you should verify that no users are connected. You must have administrator privileges to stop xfServer.

You can close a specific xfServer connection with the Monitor utility (**synxfmon.exe**). For instructions see “The Monitor Utility for Windows” in the “General Utilities” chapter of *Synergy Tools*.

To stop xfServer, do one of the following:

- Start the Synergy Configuration Program and go to the xfServer/xfServerPlus tab. Select the service from the list and click the Stop Service button.
- Go to Administrative Tools > Services. Select the service (the default display name is Synergy/DE xfServer), and then click the Stop button.

- At a command prompt enter

  ```
  net stop servicename
  ```

  where *servicename* is the name of the xfServer service to be stopped. The default service name is **SynSrv**.

- At a command prompt enter

  ```
  rsynd -q
  ```

  This stops the default xfServer, **SynSrv**, on the default port, 2330. Use the **-c** option to specify a different service name; use the **-p** option to specify a non-default port.

To both stop *and unregister* xfServer, do one of the following:

- Start the Synergy Configuration Program and go to the xfServer/xfServerPlus tab. Select the service from the list and click the Remove Service button. Click Yes at the confirmation prompt, and then click Apply.
- At a command prompt enter

  ```
  rsynd -x
  ```

  This stops and unregisters the default xfServer, **SynSrv**, on the default port, 2330. Use the **-c** option to specify a different service name; use the **-p** option to specify a non-default port.

Using xfServer on UNIX

Defining environment variables for xfServer processes

See "Why use environment variables with xfServer?" on page 3-4 for a general discussion on the benefits of using environment variables with xfServer. If your client application includes environment variables in the format LOGICAL:@*server_name*, you must define those environment variables on the server, where xfServer can find them.

Define your environment variables in the **synrc** file (in the /etc directory) and/or in the **.synrc** file (in each user's $HOME directory or in the default account's $HOME directory) on your server system. You can specify a different **synrc** file for each instance of xfServer by port number. To do this, name the file **synrc.####**, where #### is the port number, and put the file in the /etc directory. Environment variables defined in a **synrc** file for a specific port number take precedence over those set in the generic **synrc** file; environment variables set in the user's **.synrc** take precedence over both the port-specific and the generic **synrc**.

Note that the **synrc** and **synrc.####** files in the /etc directory are read when **rsynd** starts up. The **.synrc** file is read when a connection is made. Consequently, if you change values in either of the **synrc** files in the /etc directory, you must restart xfServer for the changes to take effect.

To include comments in the **synrc** file, precede the comment with a number sign (#).

1. Ensure that each client user has a valid user name and password on each server machine that is running xfServer. This user name will be used to generate the persona used during xfServer access. If you are running in non-secure mode with a default user account, only the default user account needs to have a valid user name and password on the server machine.
2. Set up a generic **synrc** file in the /etc directory of the server for all client users. If desired, set up **synrc** files for specific instances of xfServer by specifying the port number as the file extension. (A sample **synrc** file is included with your installation.)
3. Put any user-specific environment variables in another file, **.synrc**, in the user's $HOME directory on the server.

A period in the filename (**.synrc**) is used only when the file is in the user's $HOME directory. The period is *not* used when the file is in the /etc directory. (This conforms with UNIX conventions.) xfServer (**rsynd**) reads the **synrc** file only when it is *first* started. So, if you change any settings in this file, you must stop and restart **rsynd**.

Configuring xfServer

Security modes

On UNIX, xfServer can be run in trusted mode, secure mode, or non-secure mode.

Trusted mode

Trusted mode is the default. In trusted mode, the server assumes a trusted network, where users have been properly authenticated by their individual machines.

If the RUSER environment variable (or registry setting) is set on the client, it is used: the user name and encoded password (credentials) are sent to the server, where they are checked against the user's password for the user's account on the server. The absence of a password is considered a failure. If authentication fails, an error is generated and access to xfServer is denied.

If RUSER is not set, the user name of the logged-in user is sent to the server (this is referred to as implied RUSER), and the password is not checked. If the user name used to log in the client does not exist on the server, an error is generated and access to xfServer is denied.

Regardless of whether RUSER is set or implied RUSER is used, the user name sent to the server is used to generate the persona used during xfServer access.

To run xfServer in trusted mode, start **rsynd** without any mode option. To set RUSER on the client machines, run the **setruser** utility. For more information on setting RUSER, see "The setruser Utility" on page 3-66 and RUSER in the "Environment Variables" chapter of *Environment Variables & System Options*.

Secure mode

In secure mode, the client sends the user name and encoded password (credentials) to the server, where they are checked against the user's password on the user's account on the server. The absence of a password is considered a failure. If authentication fails, an error is generated and access to xfServer is denied. If authentication succeeds, the user name sent to the server is used to generate the persona used during xfServer access.

To run xfServer in secure mode, start **rsynd** with the **-s** option. You must run the **setruser** utility on each client machine. On a Windows client, **setruser** will generate the encoded password for RUSER and set RUSER in the registry. (RUSER can also be set in the environment on Windows; the environment setting takes precedence over a registry setting.) On a UNIX client, **setruser** will generate the encoded password for RUSER, which you can then use to set the RUSER environment variable.

⚠ Even though they are encoded, RUSER credentials should be kept confidential because they can be used with any xfServer client.

For more information on setting RUSER, see "The setruser Utility" on page 3-66 and RUSER in the "Environment Variables" chapter of *Environment Variables & System Options*.

Non-secure mode

In non-secure mode, if RUSER is set on the client, the user name is sent to the server; otherwise, the user name of the logged-in user is sent. No password is checked. If the user doesn't have an account on the server, an error is generated and access to xfServer is denied. If the user does have an account on the server, it is used to generate the persona used during xfServer access.

To run xfServer in non-secure mode, start **rsynd** with the **-n** option.

When running in non-secure mode, you have the option of specifying a default user account (with the **-u** option), which will be used as the persona for all clients. (See the examples below for the syntax.) If you choose to use a default user account, you do not need to set RUSER on the clients (if it is set, it is ignored) nor create an account for each client on the server. To use the default account option, we recommend that you create an account on the server with limited privileges (this account cannot be root) specifically for use with xfServer. **Rsynd** will validate that the specified account is a valid account on the local machine before starting the service. If you are going to use non-secure mode, we recommend that you use a default user account.

If you are running xfServer in non-secure mode with encryption enabled, you *must* specify a default user account.

There are two ways to start **rsynd** when using the default account option—with a password and without a password.

- This method requires that you supply the password for the default user account when starting **rsynd**. The password must be encoded with **setruser**; a clear text password is invalid. There are a couple of ways to do this. For example:

  ```
  rsynd -n -u `setruser`
  ```

  This command launches the **setruser** utility, which prompts for a user name and password. After you enter the user name and password, the command starts **rsynd** with the specified user name and the encoded password. (Note that **setruser** must be enclosed between grave accent characters (`) as shown above.)

  You can also run **setruser** to generate the encoded username/password string, and then include it in the start-up command, as shown in this example:

  ```
  rsynd -n -u "username/\362\224c\261\351\224\374P"
  ```

  Because this method does not require user input, you can put this command in a start-up file. Note that the username/password string must be enclosed in double quotation marks; failure to include the quotation marks may result in a "wrong username/password" error. For information on using **setruser**, see "The setruser Utility" on page 3-66.

- This method enables you to start **rsynd** from an authorized account without specifying a password:

  `rsynd -n -u` *xfServerAccount*

  where *xfServerAccount* is the user name of the default user account you created for use with xfServer. This command can be executed by a user signed on as *xfServerAccount* or by any user with root privileges (uid=0). A password is not required; if it is passed, it is ignored.

Logging

By default, xfServer logs start and stop activity and errors to the file **/usr/lib/rsynd.log**. Use the environment variable RSYNDLOG to specify an alternate log file name. By default, the maximum log file size is 10K. Use the RSLOGMAX environment variable to specify a larger log file size. Set RSYNDLOG and RSLOGMAX as described in "Defining environment variables for xfServer processes" on page 3-30. For more information, see RSYNDLOG and RSLOGMAX in the "Environment Variables" chapter of *Environment Variables & System Options*.

Ports

The default port used by xfServer is 2330. If you are running multiple instances of xfServer on the same machine, each will need to run on a different port. To specify a non-default port, start **rsynd** with the **-p** option and set the SCSPORT environment variable on the client. A non-default port must be specified on both the server and the client. For more information, see SCSPORT in the "Environment Variables" chapter of *Environment Variables & System Options*.

Compression

Set the SCSCOMPR environment variable to a non-zero value in the **synrc** file on the server (see "Defining environment variables for xfServer processes" on page 3-30) to compress data records sent between xfServer and its clients. This option compresses blanks, nulls, zeros, and repeating characters. Compression can significantly improve performance on low speed or busy networks, especially WANs.

SCSCOMPR can also be set on the client or in the client's **.synrc** file on the server. To turn compression off, it must be turned off on both the server and the client. For more information, see SCSCOMPR in the "Environment Variables" chapter of *Environment Variables & System Options*.

Starting xfServer

The xfServer daemon (**rsynd**) is a background process that must be running for xfServer to work.

1. To enable the xfServer daemon to grant adequate resources to clients, verify that the **rsynd** program is owned by root. In the dbl/bin directory, enter

   ```
   chown root rsynd
   ```

2. Make sure that **rsynd** has the set-user-ID bit set. In the same location, enter

   ```
   chmod u+s rsynd
   ```

3. Add the full path name of **rsynd** to the start-up file (the run command file, **rc**) on your server system. (The actual file name may vary depending on your operating system. See your system administrator or operating system documentation.) This will cause xfServer to start and be ready for incoming client application requests each time the server is booted.

   For example, on some machines you might add this line to the **/etc/rc** file:

   ```
   /usr/synergyde/dbl/bin/rsynd
   ```

Stopping xfServer

There are two ways to stop the xfServer daemon (**rsynd**).

- To stop the xfServer daemon without killing the existing connections, use the **-q** option:

  ```
  rsynd -q
  ```

  This stops xfServer on the default port, 2330. If you are using a different port, specify the port number with the **-p** option.

  This is the normal method for stopping **rsynd**. Existing connections are allowed to continue, but new connections are blocked. Use this method when you need to start a new version or configuration of **rsynd** on that port, or any time you want to prevent new access to the server without interrupting existing connections.

- To stop the xfServer daemon *and kill all existing connections*, use the **-c** option with **-q**:

  ```
  rsynd -q -c
  ```

  This stops xfServer on the default port, 2330. If you are using a different port, specify the port number with the **-p** option.

  All existing connections are terminated and new connections are blocked. Use this method only when you need exclusive access to the server data.

  ⚠ Use this method with caution. Make sure you fully understand the effects of **-q -c** before stopping **rsynd** in this way.

When you use **-q -c**, the client will be unaware that the server has been terminated until it attempts to access data on a channel that has been dropped. At this point, the client will receive a “Network problem reaching server %s” error ($ERR_NETPROB). If you later want to reconnect to the server without terminating the client application, all channels currently open to the server must first be closed. While the server is down, any attempt to open a new channel will receive a “Synergy server is not running on %s” error ($ERR_NOSERVER), which can be trapped. Once the server has been restarted, a new open channel will reconnect to the server.

Using xfServer on OpenVMS

Defining logical names for xfServer processes

See "Why use environment variables with xfServer?" on page 3-4 for a general discussion on the benefits of using logicals with xfServer. If your client application includes environment variables in the format LOGICAL:@*server_name*, you must define those environment variables on the server, where xfServer can find them.

Logicals can be defined at different levels using a number of different methods. It's important to understand the hierarchy and implications of each method, as well as how they can be used together. (See Examples on page 3-39 for an example showing how two of the methods can be used together.) The information in this section applies to both xfServer and xfServerPlus.

Understanding tables

The sections below describe methods that put logicals in the system table, the LNM$RSDMS$MGR_*port* table (where *port* is the port number **rsynd** is listening on), and the LNM$SYNSVR_*processID* table (where *processID* is the ID of an xfServer process).

The LNM$RSDMS$MGR_*port* and LNM$SYNSVR_*processID* tables are created when **rsynd** starts. The LNM$RSDMS$MGR_*port* table is shared by all pooled processes for a single instance of **rsynd**. If you run **rsynd** on multiple ports, there will be a separate table for each port (for example, LNM$RSDMS$MGR_2330, LNM$RSDMS$MGR_2440). The LNM$SYNSVR_*processID* table is specific to each xfServer process.

Tables are scanned by the session servers in this order: process table, LNM$SYNSVR_*processID*, LNM$RSDMS$MGR_*port*, job table, group table, system table. Once a logical is found in a table, the remaining tables are not searched. For example, if the logical DATA is defined in both the LNM$RSDMS$MGR_*port* table and the system table, the definition in the LNM$RSDMS$MGR_*port* table will be used.

Defining system-level logicals

You can define logicals intended for system-wide visibility in the **SERVER_INIT.COM** file. The **SERVER_INIT.COM** file is invoked in the **SYNERGY_STARTUP.COM** file after the connection manager daemon (**rsynd**) has been started. The logicals are placed in the system table. Consequently, all instances of xfServer will use the same logical definitions, and other applications on the system will also have access to these definitions.

The **SERVER_INIT.COM** file may contain any DCL command, so you can use this method for other xfServer-related system start-up commands. Because the **SERVER_INIT.COM** file is processed by DCL, you can use this method to put logicals into other tables at system start-up.

To use this method, create a file named **SERVER_INIT.COM** in DBLDIR and use this qualifier to define the logicals:

```
$ DEFINE/SYS logical_name device:[directory]
```

For example:

```
$ DEFINE/SYS DATA DKA600:[SYNERGYDE.CONNECT]
```

If you use this method for defining logicals, you should edit the **SYNERGY_STARTUP.COM** file to start the server with the /NOUSE_SYNRC qualifier. This tells xfServer to ignore the **SYNRC.COM** file in the home directory of the user when setting up logicals. See "The rsynd Program" on page 3-44 for complete information on **rsynd** qualifiers.

Defining logicals for use by all instances of xfServer

If you have several instances of xfServer running on different ports and would like them all to use the same logical definitions (but you don't want the logicals available to all applications on the system), you can define the logicals in the **DBLDIR:SYNRC.COM** file. (Note that logicals defined in **SYNRC.COM** are case sensitive.)

Rsynd processes the **DBLDIR:SYNRC.COM** file when it starts up and places the logicals in the LNM$RSDMS$MGR_*port* table. They are then available to all session servers started by **rsynd**. Although the logicals are placed in the LNM$RSDMS$MGR_*port* table, all instances of xfServer will use the same logicals because this method does not enable you to specify a port. If you have multiple instances of xfServer, there will be multiple LNM$RSDMS$MGR_*port* tables, but the logical definitions will be the same for all of them.

Because **DBLDIR:SYNRC.COM** is opened and interpreted by **rsynd** (rather than DCL), you cannot place other DCL commands in this file, and any table qualifiers placed on the DEFINE command are ignored. When using this method, **rsynd** start-up is slower than normal because it must process **SYNRC.COM**.

To use this method, create a file named **SYNRC.COM** in DBLDIR and use this qualifier to define the logicals:

```
$ DEFINE logical_name device:[directory]
```

For example:

```
$ DEFINE DATA DKA600:[SYNERGYDE.CONNECT]
```

If you use this method for defining logicals, you should edit the **SYNERGY_STARTUP.COM** file to start the server with the /NOUSE_SYNRC qualifier. This tells xfServer to ignore the **SYNRC.COM** file in the home directory of the user when setting up logicals. See "The rsynd Program" on page 3-44 for complete information on **rsynd** qualifiers.

Defining server-specific logicals

You can define server-specific logicals in the **SERVER_INIT.COM** file. The **SERVER_INIT.COM** file is invoked in the **SYNERGY_STARTUP.COM** file after the connection manager daemon (**rsynd**) has been started. The logicals are placed in the LNM$RSDMS$MGR_*port* table.

This method allows you to specify a port for xfServer, which enables you to define logicals differently for separate instances of xfServer. **SERVER_INIT.COM** places logicals in the LNM$RSDMS$MGR_*port* table *after* **DBLDIR:SYNRC.COM** has already done so. Consequently, the logicals defined in **SERVER_INIT.COM** will override the same logicals defined in **DBLDIR:SYNRC.COM**.

The **SERVER_INIT.COM** file may contain any DCL command, so you can use this method for other xfServer-related system start-up commands. Because the **SERVER_INIT.COM** file is processed by DCL, you can use this method to put logicals into other tables at system start-up.

To use this method, create a file named **SERVER_INIT.COM** in DBLDIR and use this qualifier to define the logicals:

```
$ DEFINE/TABLE=LNM$RSDMS$MGR_port/USER logical_name device:[directory]
```

where *port* is the port number on which xfServer is listening. For example:

```
$ DEFINE/TABLE=LNM$RSDMS$MGR_2330/USER DATA DKA600:[SYNERGYDE.CONNECT]
```

If you use this method for defining logicals, you should edit the **SYNERGY_STARTUP.COM** file to start the server with the /NOUSE_SYNRC qualifier. This tells xfServer to ignore the **SYNRC.COM** file in the home directory of the user when setting up logicals. See "The rsynd Program" on page 3-44 for complete information on **rsynd** qualifiers.

Defining user-specific logicals

You can define user-specific logicals by placing them in the **SYNRC.COM** file in the home directory of the user. When a remote user logs in to a session server, that session server processes **SYNRC.COM** in the user's home directory (SYS$LOGIN) and places the logicals in the LNM$SYNSVR_*processID* table. This table is read *before* the LNM$RSDMS$MGR_*port* table and the system table, so logicals defined in this manner override logicals defined with any of the other methods discussed above.

When you run xfServer in non-secure mode with a default user account, a single account is used as the persona for all users. Consequently, you cannot specify different logicals for individual users. This also applies to xfServerPlus because it uses a single account to run all xfServerPlus sessions.

To use this method, create a file named **SYNRC.COM** in the home directory of the user who is connecting to the server, and use this qualifier to define the logicals:

$ DEFINE *logical_name device*:[*directory*]

For example:

```
$ DEFINE DATA DKA600:[BETTY]
```

Examples

This example shows you how to use the methods described in "Defining logicals for use by all instances of xfServer" and "Defining server-specific logicals" together. Say you have an accounting application with data for two customers, and the directory structure for the data is the same for both customers (only the name of the main directory differs). Within the main directory for each customer are subdirectories for AP, AR, and GL. The system consists of Windows clients connecting to an OpenVMS server.

You could set up logicals in the **SERVER_INIT.COM** file to access the data on two ports like this:

```
$ DEFINE/TABLE=LNM$RSDMS$MGR_2330/USER/TRANS=CONC CUST DKA600:[CUST_ONE.]
$ DEFINE/TABLE=LNM$RSDMS$MGR_2331/USER/TRANS=CONC CUST DKA600:[CUST_TWO.]
```

Users who connect on port 2330 will have access to the data files for customer one, while users who connect on 2331 will have access to the data files for customer two.

Then, in the **DBLDIR:SYNRC.COM** file, set up logicals for the subdirectories of CUST_ONE and CUST_TWO. These will apply to all **rsynd**s. For example:

```
$ DEFINE AP CUST:[AP]
$ DEFINE AR CUST:[AR]
$ DEFINE GL CUST:[GL]
```

On the Windows client, set SCSPORT to either 2330 or 2331, and set up logicals for AP, AR, and GL that point to the server. For example:

```
set SCSPORT=2331
set MY_AP=AP:@vms_server
set MY_AR=AR:@vms_server
set MY_GL=GL:@vms_server
```

Within the Synergy application, there is code to open the file named **MY_AP:ap.ism**. As defined above, this logical indicates xfServer access and will attempt to open **AP:ap.ism** on the server. On the OpenVMS server, it is translated as **CUST:[AP]AP.ISM**. Then, because the client connected on port 2331, this is translated as **DKA600:[CUST_TWO.AP]AP.ISM**, and the file is opened.

Configuring xfServer

Security modes

On OpenVMS, xfServer can be run in secure mode or non-secure mode. In both secure and non-secure mode, each client user must have a valid user name and password on each server machine that is running xfServer. If you are running in non-secure mode with a default user account, only the default user account needs to have a user name and password on the server machine.

Secure mode

In secure mode, the client sends the user name and encoded password (credentials) to the server, where they are checked against the user's password on the user's account on the server. The absence of a password is considered a failure. If authentication fails, an error is generated and access to xfServer is denied. If authentication succeeds, the user name is used to generate the persona used during xfServer access.

To start xfServer in secure mode, include the /SECURE option on the command line in the **SYNERGY_STARTUP.COM** file. You must run the **setruser** utility on each client. On a Windows client, **setruser** will generate the encoded password for RUSER and set RUSER in the registry. (RUSER can also be set in the environment on Windows; the environment setting takes precedence over a registry setting.) On a UNIX client, **setruser** will generate the encoded password for RUSER, which you can then use to set the RUSER environment variable. For more information on setting RUSER, see "The setruser Utility" on page 3-66 and RUSER in the "Environment Variables" chapter of *Environment Variables & System Options*.

⚠ Even though they are encoded, RUSER credentials should be kept confidential because they can be used with any xfServer client.

Non-secure mode

Non-secure mode is the default. If the RUSER environment variable (or registry setting on a Windows client) is set on the client, the user name is sent to the server; otherwise, the user name of the logged-in user is sent (this is referred to as implied RUSER). No password is checked. If the user doesn't have an account on the server, an error is generated and access to xfServer is denied. If the user does have an account on the server, it is used to generate the persona used during xfServer access.

To start xfServer in non-secure mode, you can specify /NOSECURE on the command line in the **SYNERGY_STARTUP.COM** file if you like, but it is not required since non-secure is the default.

When running in non-secure mode, you have the option of specifying a default user account (with the /DEFAULT_USER option), which will be used as the persona for all clients. If you choose to use a default user account, you do not need to set RUSER on the clients (if it is set, it is ignored) nor create an account for each client on the server. To use the default account option, we recommend that you create an account on the server with limited privileges specifically for use with xfServer.

Rsynd will validate that the specified account is a valid account on the local machine before starting the process. This account cannot have the privileges listed below unless the /ALLOW_PRIVILEGED qualifier is specified in the start-up command. We recommend that you *not* allow these privileges because they give end users privileges at the system administrator level.

ALTPRI
BYPASS
CMEXEC
CMKRNL
DETACH
DOWNGRADE
EXQUOTA
READALL
SECURITY
SETPRV
SYSNAM
SYSPRV
VOLPRO
WORLD

If you are running xfServer in non-secure mode with encryption enabled, you *must* specify a default user account.

To start xfServer in non-secure mode with a default user, include the following options on the command line in the **SYNERGY_STARTUP.COM** file:

```
/NOSECURE /DEFAULT_USER=user_name
```

where *user_name* is the account that you created to use with xfServer. Note that you *must* specify the /NOSECURE option on the command line when you are using a default user account.

Logging

By default, logging of rsynd errors and some status messages is enabled (/LOG_LEVEL=ENABLE). Because this level of logging is a program default, it is not necessary to include this qualifier on the start-up command line in the **SYNERGY_STARTUP.COM** file. To send these messages to a log file, use the /OUTPUT qualifier on the start-up command line and specify a log file name. By default, messages are logged to the file ***node*_rsynd_*port*.log** in DBLDIR.

To log additional status information and start/stop messages, use the /LOG_LEVEL=FULL qualifier on the start-up command line in conjunction with /OUTPUT.

To send messages to the operator console in addition to the log file, use the /LOG_LEVEL=OPERATOR qualifier on the start-up command line and set REPLY/ENABLE=NETWORK on your operator terminal. You can use the OPERATOR setting with either regular or full logging. For example, if you want to enable full logging and send messages to the operator console, use /LOG_LEVEL=(FULL, OPERATOR).

To turn logging off completely, use /LOG_LEVEL=NONE.

To enable logging of data regarding clients that connect to the OpenVMS server, use /CLIENT_LOGFILE. This creates a file named ***node*_SRVR_*pid*.log** in DBLDIR. If /CLIENT_LOGFILE is not set (or /NOCLIENT_LOGFILE *is* set), client logs are created only when there is an error.

Ports

The default port for xfServer is 2330. If you are running multiple instances of xfServer on the same machine, each will need to run on a different port. To start xfServer on a non-default port, in the **SYNERGY_STARTUP.COM** command file specify the following option:

```
/PORT=nnnn
```

A non-default port must be specified on both the server and the client. Use the SCSPORT environment variable on the client system to specify the port for xfServer. (For a Windows client, set SCSPORT in the registry.) For more information, see SCSPORT in the “Environment Variables” chapter of *Environment Variables & System Options*.

Compression

Set the SCSCOMPR logical to a non-zero value in the **DBLDIR:SYNRC.COM** or **SERVER_INIT.COM** file (see “Defining server-specific logicals” on page 3-38) to compress data records sent between xfServer and its clients. This option compresses blanks, nulls, zeros, and repeating characters. Compression can significantly improve performance on low speed or busy networks, especially WANs.

SCSCOMPR can also be set on the client. To turn compression off, it must be turned off on both the server and the client. For more information, see SCSCOMPR in the “Environment Variables” chapter of *Environment Variables & System Options*.

Starting xfServer

xfServer is normally started automatically at machine startup using the **SYNERGY_STARTUP.COM** file. The **rsynd** startup command is placed in this file when Synergy DBL is installed.

To start xfServer manually, execute the following command with the desired switches while logged into the SYSTEM account. (See “The rsynd Program” on page 3-44 for the complete **rsynd** syntax.)

```
$ RSYND
```

To start xfServer manually using the switches defined in the **SYNERGY_STARTUP.COM** file, execute the following command while logged into the SYSTEM account:

```
$ @SYNERGY_STARTUP XFRESTART
```

xfServer can also be started from the **servstat** program. See “The servstat Program” in the “General Utilities” chapter of *Synergy Tools* for more information.

On OpenVMS, xfServer sets the current working directory to the user’s default directory and creates a SYS$SCRATCH logical to reference the user’s directory.

Stopping xfServer

There are several ways to stop xfServer (**rsynd**):

- To stop xfServer without shutting down existing connections, execute the following command while logged into the SYSTEM account:

  ```
  $ RSYND/SHUTDOWN/PORT=nnnn
  ```

  where *nnnn* specifies the port number of the server system to shut down. After this command has been successfully executed, existing connections are allowed to continue, but new connections are blocked.

- To stop xfServer *and shut down all existing connections*, execute the following command while logged into the SYSTEM account:

  ```
  $ RSYND/SHUTDOWN=ALL/PORT=nnnn
  ```

  where *nnnn* specifies the port number of the server system to shut down. After this command has been successfully executed, all existing connections are terminated and new connections are blocked.

You can also use the **servstat** program to shut down xfServer. See "The servstat Program" in the "General Utilities" chapter of *Synergy Tools* for more information.

The rsynd Program

The **rsynd** program is used by both xfServer and xfServerPlus. (Using the **-w** option indicates that commands apply to xfServerPlus; for additional information about xfServerPlus, see the *xfNetLink & xfServerPlus User's Guide*.)

- On Windows, xfServer runs as the **rsynd** service **SynSrv** by default, and xfServerPlus runs as **xfspl** by default; see "Windows options" below.
- On UNIX, xfServer and xfServerPlus run as the **rsynd** daemon; see "UNIX options" on page 3-50.
- On OpenVMS, xfServer and xfServerPlus run as the **rsynd** detached process; see "OpenVMS options" on page 3-54.

Syntax

`rsynd` *[-option] [...]*

Windows options

⚠ Do not attempt to issue **rsynd** commands from the command line while the Synergy Configuration Program is running.

-a

Allow persona that is a member of the administrators group as the username/password for the default user account when running xfServer in non-secure mode or as the username/password under which to run xfServerPlus sessions. We do not recommend that you use this option in a production environment, as it gives remote users administrator privileges on your server. Used with **-r** (or **-rs**), **-n**, and **-u** for xfServer; used with **-r** (or **-rs**) and **-w** for xfServerPlus.

-c*name*

Specify that this **rsynd** request is only for the xfServer or xfServerPlus service named *name*. (The service name appears as a registry key in the Windows registry and is used to start and stop xfServer/xfServerPlus from the command line.) The default service name for xfServer is **SynSrv**; for xfServerPlus, it is **xfspl**. Used with **-q**, **-r**, **-rs**, or **-x**.

To run multiple instances of xfServer or xfServerPlus, each must have a different service name, display name, and port number. See example A on page 3-48.

-cert=*filename*

Specify the digital certificate file (**.pem** file) to be used for data packet encryption. The file may not require a pass phrase. The default filename and location is **DBLDIR:rsynd.pem**, which will be used if **-cert** is not specified. *Filename* may use a complete path or a logical. Used with **-cipher**, **-encrypt**, and **-scl**.

-cipher=*level*

Specify the desired level of encryption. Valid values for *level* are HIGH, MEDIUM, LOW. If not specified, MEDIUM is used. These values map to specific cipher suite names and protocols, which are determined by the version of OpenSSL being used. If MEDIUM or LOW is specified, the security compliance level (**-scl**) must be **1**. You can use the SLC option to the GETFA routine to see which cipher suite and protocol is being used. See GETFA in the "System-Supplied Subroutines and Functions" chapter of the *Synergy DBL Language Reference Manual*. Used with **-cert**, **-encrypt**, and **-scl**.

-d*name*

Specify a display name to be associated with the service name. (The display name appears in the Services dialog box.) The default display name for xfServer is **Synergy/DE xfServer**; for xfServerPlus, it is **Synergy/DE xfServerPlus**. Enclose the name in quotes if it contains a space. Used with **-r** or **-rs**.

-encrypt*[=level]*

Enable data packet encryption. Valid values for *level* are MASTER and SLAVE. If *level* is not specified, slave encryption is used by default. Used with **-cert**, **-cipher**, and **-scl**. See "Using Client/Server Encryption" on page 3-7 for details on xfServer encryption or "Using Encryption" in the "Configuring and Running xfServerPlus" chapter of the *xfNetLink & xfServerPlus User's Guide* for details on xfServerPlus encryption.

-h

Display a list of **rsynd** options.

-keepalive=*time*

(xfServer only) Sets the value for the TCP keepalive timer in seconds for xfServer connection recovery. Valid values are 1 through 7200 (2 hours). If not specified, defaults to 150 seconds (2.5 minutes). Used with **-keepconnect**, **-keepcontext**, **-keeplocks**, and **-retrytime**. See "Using connection recovery" on page 3-19 for details on this feature. (Note that this option applies *only* when connection recovery is enabled. See "Rsynd: shutting down and timing out" on page 3-71 for general information about keepalive.)

-keepconnect=*level*

(xfServer only) Specify the level for connection recovery. Valid values are OFF, MASTER, and SLAVE. By default, this feature is enabled in slave mode. To disable the feature, use **-keepconnect=OFF**. Used with **-keepalive**, **-keepcontext**, **-keeplocks**, and **-retrytime**. See "Using connection recovery" on page 3-19 for details on this feature.

-keepcontext=*time*

(xfServer only) The amount of time, in seconds, that xfServer will maintain a client context after an unexpected socket disconnect has been detected. Valid values are 1 through 260000 (72 hours). If not specified, defaults to 300 seconds (5 minutes). Used with **-keepalive**, **-keepconnect**, **-keeplocks**, and **-retrytime**. See "Using connection recovery" on page 3-19 for details on this feature.

-keeplocks=*time*

(xfServer only) The amount of time, in seconds, that xfServer will maintain file and record locks for a client context after an unexpected socket disconnect has been detected. Valid values are 0 through 43200 (12 hours). If not specified, defaults to 300 seconds (5 minutes). Set to 0 if you want locks to be released as soon as a disconnected socket is detected. This value must be smaller than the **-keepcontext** value. Used with **-keepalive**, **-keepconnect**, **-keepcontext**, and **-retrytime**. See "Using connection recovery" on page 3-19 for details on this feature.

-n

(xfServer only) Run xfServer in non-secure mode. Used with **-r** or **-rs**. See the **-u** option to specify a default user account for anonymous data access in non-secure mode.

-p*port*

Specify the port on which xfServer or xfServerPlus listens. The default port is 2330 for xfServer and 2356 for xfServerPlus. Valid ports are in the range 1024 through 65535. Used with **-q**, **-r**, **-rs**, or **-x**. *Port* updates the SCSPORT registry entry for the service.

-q

Stop the xfServer or xfServerPlus service specified with **-c**. If **-c** is not specified, either **SynSrv** is stopped or—if **-w** is specified—**xfspl** is stopped. If the service is running on a non-default port, specify the port number with **-p**. Existing connections are lost. See also the **-x** option.

-r

Register (install) a new service using the name specified by **-c**. If **-c** is not specified, either **SynSrv** is registered or—if **-w** is specified—**xfspl** is registered.

-rd*port[***:***timeout]*

(xfServerPlus only) Enable remote debugging via Telnet for xfServerPlus on the specified port. Used with **-r** (or **-rs**) and **-w**. xfServerPlus will listen on the specified port for the Telnet client; this port number must be different than the port that xfServerPlus is running on (2356 or the port specified with **-p**). If desired, specify a time-out in seconds; this is the length of time that the server will wait for a connection from the Telnet client after the xfNetLink–xfServerPlus connection has been made. The default is 100 seconds. See example D on page 3-49.

-retrytime=*time*

(xfServer only) The amount of time, in seconds, that a client will continue to attempt to reconnect to an xfServer thread that has been unexpectedly disconnected when connection recovery is enabled. Valid values are 1 through 600. If not specified, defaults to 60 seconds. The client will issue a reconnect every 5 seconds for *time* seconds before returning a "Network problem reaching server" ($ERR_NETPROB) error. Used with **-keepalive**, **-keepconnect**, **-keepcontext**, and **-keeplocks**. See "Using connection recovery" on page 3-19 for details on this feature.

-rs

Register a new service using the name specified by **-c** and then start it. If **-c** is not specified, either **SynSrv** is registered and started or—if **-w** is specified—**xfspl** is registered and started.

-s

(xfServer only) Run xfServer in secure mode. This is the default. Used with **-r** or **-rs**.

-scl*[=level]*

Specify the security compliance level (that is, the available protocols). Valid values are 0, 1, and 2.

0 Use default protocols for current version of Synergy
1 Use protocols TLS1.0, TLS1.1, TLS1.2 (current default)
2 Use protocols TLS1.1, TLS1.2

If **-scl** is not specified, or if it's specified without *level*, **0** is used. A value of **0** means that *the available protocols may change with the version of Synergy*; the current default is indicated above. If you don't want the available protocols to change when you upgrade xfServer, specify level **1** or **2**. Level 2 requires that **-cipher** be set to HIGH. Used with **-cert**, **-cipher**, and **-encrypt**. See "Understanding cipher suites and protocols" on page 3-8 for more information.

-sspi

(xfServer only) Run xfServer in restricted security mode (Windows authentication only). See "Understanding xfServer security" on page 3-16 for more information. Used with **-r** or **-rs**.

-text *"description"*

(xfServerPlus only) Add the string specified in *description* to the end of the xfServerPlus command line. Once the **dbs** process starts (which happens only after an xfNetLink client makes a connection), this string displays at the end of the command line for the **dbs** process in the Command Line column of the Processes tab in Windows Task Manager. (You may need to add the Command Line column to the Processes display.) If you have several xfServerPlus processes running at once, this enables you to distinguish among them. If *description* contains %s, it is replaced with the IP address of the xfNetLink client. (Other *%letter* variables should not be used.) Enclose the string in quotation marks, but do not include quotation marks *within* the string.

-u*persona*

xfServer: Specify *persona* as the default user name and password when running xfServer in non-secure mode. Separate the user name and password with a forward slash. The password must be clear text. The account can be on the local machine or it can be a domain controller account. If the latter, use either *user_name@domain_name/password* or *domain_name\user_name/password*. Used with **-r** (or **-rs**) and **-n**. This option is required when **-encrypt** is used in non-secure mode. See example B on page 3-48.

xfServerPlus: Specify *persona* as the user name and password under which to run xfServerPlus sessions. Separate the user name and password with a forward slash. The password must be clear text. The account can be on the local machine or it can be a domain controller account. If the latter, use either *user_name@domain_name/password* or *domain_name\user_name/password*. Used with **-r** (or **-rs**) and **-w**. The **-u** option is required for xfServerPlus.

-v

Display the **rsynd** version.

-v6

Server has an IPv6 address. Default is IPv4. Specifying **-v6** means both IPv4 and IPv6 are supported. This option can be changed only from the command line; it is not currently available in the Synergy Configuration Program.

-w

(xfServerPlus only) Specify that this **rsynd** request is for xfServerPlus (rather than xfServer). Used with **-q**, **-r**, **-rs**, or **-x**. See the *xfNetLink & xfServerPlus User's Guide* for information on xfServerPlus.

-x

Unregister (remove) the service specified with **-c**. If **-c** is not specified, either the default service, **SynSrv,** is unregistered or—if **-w** is specified—**xfspl** is unregistered. If the service is running on a non-default port, specify the port number with **-p**. If the service is running, **-x** will first stop it, which will cause any existing connections to be lost, and then unregister it. See also the **-q** option.

Windows usage

For options that do not include an equals sign, you can place a space between an option and its argument if desired. When an equals sign is shown in the syntax, it is required.

A. The example below shows how to register **rsynd** for xfServer and specify a non-default port (2445; the default is 2330), service name (MyServer), and display name (My Server on 2445). The *service name* (also referred to as the registry key) is the name you see in the Windows registry. The *display name* is the name you see in the Component Services dialog box when you start and stop xfServer. This xfServer will run in secure mode (if no mode is specified, the default is secure).

```
rsynd -p 2445 -r -c MyServer -d "My Server on 2445"
```

B. The following example shows how to register **rsynd** for xfServer and run in non-secure mode (**-n**) with a default user account (**-u**). With the **-u** option, we have specified the user name and password for the account that we created for use with xfServer. The default port (2330), service name (SynSrv), and display name (Synergy/DE xfServer) will be used.

```
rsynd -r -n -u Tiger/Mickey123*
```

C. The following example shows how to register and start (**-rs**) **rsynd** for xfServerPlus. Because we did not specify a port, xfServerPlus will run on the default port (2356). The default service name (**xfspl**) and display name (**Synergy/DE xfServerPlus**) will be used.

```
rsynd -rs -w -u xfspAcct/Minnie123*
```

D. The following example shows how to register **rsynd** for xfServerPlus with remote debugging via Telnet enabled. We have specified 2445 as the debug port number (the port the server will listen on for the Telnet client) and 60 seconds as the time-out value. Note that no spaces are allowed between the port number, colon, and time-out value. xfServerPlus will use the default port (2356), service name, and display name. For more information about remote debugging via Telnet, see "Debugging Your Remote Synergy Routines" in the "Configuring and Running xfServerPlus" chapter of the *xfNetLink & xfServerPlus User's Guide*.

```
rsynd -r -rd 2445:60 -w -u xfspAcct/Minnie123*
```

E. The following example shows how to register **rsynd** for xfServer and enable slave encryption using defaults for the cipher, **.pem** filename, security compliance level. The service will run in secure mode and use the default service name and display name.

```
rsynd -r -p 2445 -encrypt
```

F. The following example shows how to register **rsynd** for xfServer and enable master encryption and specify values for the other encryption options.

```
rsynd -r -p 2445 -encrypt=master -cipher=high -cert=CERT:myFile.pem -scl=2
```

G. Note that if you want slave encryption, and you have used either the **-cipher** or the **-cert** option, you are not required to include the **-encrypt** option. For example, the following command will register xfServerPlus and enable slave encryption using the default **.pem** filename and default security compliance level.

```
rsynd -r -w -u xfspAcct/Minnie123* -p 2445 -cipher=high
```

H. The following example shows how to register **rsynd** for xfServer and enable master connection recovery and specify values for KeepContext and KeepLocks. Defaults will be used for the keepalive and retrytime options.

```
rsynd -r -p 2445 -keepconnect=master -keepcontext=600 -keeplocks=600
```

I. Note that if you want slave connection recovery, and you have used any of the related time options (**-keepalive**, **-keepcontext**, **-keeplocks**, **-retrytime**), you are not required to include the **-keepconnect** option. For example, the following command will register xfServer and enable slave connection recovery using default settings for all the times except retrytime:

```
rsynd -r -p 2445 -retrytime=90
```

The above examples show only how to *register* the service. You'll still need to start it from the command line, the Synergy Configuration Program, or the Windows Services console. See "Starting xfServer" on page 3-27 for details. If you prefer, you can use the **-rs** option to register and start **rsynd** in one step.

J. The following example shows how to stop the MyServer service running on port 2445:

```
rsynd -c MyServer -p 2445 -q
```

UNIX options

<none>

Start xfServer. If a mode is not specified with **-n** or **-s**, xfServer runs in trusted mode. See “Security modes” on page 3-31 for more information.

-a

Allow super user as the persona of the default user account or as the client for xfServer, or allow super user as the persona under which to run xfServerPlus sessions. We do not recommend that you use this option in a production environment, as it gives remote users root access on your server. Used with **-n** and **-u** for xfServer; used with **-w** for xfServerPlus.

-c

Stop xfServer or xfServerPlus and kill all existing connections. Used with **-q**. If **-p** is not specified, the xfServer process on port 2330 will be stopped or—if **-w** is specified—the xfServerPlus process on 2356 will be stopped. See also the **-q** option.

See also “Stopping xfServer” on page 3-34. For more information on stopping xfServerPlus, see “Running xfServerPlus on UNIX” in the “Configuring and Running xfServerPlus” chapter of the *xfNetLink & xfServerPlus User’s Guide*.

-cert=*filename*

Specify the digital certificate file (SSL **.pem** file) to be used for data packet encryption. The file may require a pass phrase; if it does, the message “Enter PEM pass phrase” will be displayed on start-up. The default filename and location is **DBLDIR:rsynd.pem**, which will be used if **-cert** is not specified. *Filename* may use a complete path or a logical. Used with **-cipher**, **-encrypt**, and **-scl**.

-cipher=*level*

Specify the desired level of encryption. Valid values for *level* are HIGH, MEDIUM, LOW. If not specified, MEDIUM will be used. These values map to specific cipher suite names and protocols, which are determined by the version of OpenSSL being used. If MEDIUM or LOW are specified, **-scl** must be set to **1**. You can use the SLC option to the GETFA routine to see which cipher suite and protocol is being used. See GETFA in the “System-Supplied Subroutines and Functions” chapter of the *Synergy DBL Language Reference Manual*. Used with **-cert**,**-encrypt**, and **-scl**.

-encrypt*[=level]*

Enable data packet encryption for xfServer or—if **-w** is specified—for xfServerPlus. Valid values for *level* are MASTER and SLAVE. If *level* is not specified, slave encryption is used. Used with **-cert**, **-cipher**, and **-scl**. See “Using Client/Server Encryption” on page 3-7 for details on xfServer encryption or “Using Encryption” in the “Configuring and Running xfServerPlus" chapter of the *xfNetLink & xfServerPlus User's Guide* for details on xfServerPlus encryption.

-h

Display a list of **rsynd** options.

-i*address*

(xfServer only) Bind the specified IP address to xfServer. Used when the machine has more than one IP address.

-l*filename*

(xfServer only) Specify the monitor log filename. Used with **-m**.

-m

(xfServer only) Start the monitor utility for xfServer. See "The Monitor Utility for UNIX" in the "General Utilities" chapter of *Synergy Tools* for information about running the monitor.

-n

(xfServer only) Run xfServer in non-secure mode. See also the **-u** option.

-p*port*

Specify the port on which xfServer or xfServerPlus listens. The default port is 2330 for xfServer and 2356 for xfServerPlus. *Port* overrides the SCSPORT environment variable.

-q

Stop xfServer or xfServerPlus without killing existing connections. If **-p** is not specified, the xfServer process on port 2330 will be stopped or—if **-w** is specified—the xfServerPlus process on 2356 will be stopped. See also the **-c** option.

-rd*port[***:***timeout]*

(xfServerPlus only) Enable remote debugging via Telnet for xfServerPlus on the specified port. Used with **-w**. xfServerPlus will listen on the specified port for the Telnet client; this port number must be different than the port that xfServerPlus is running on (2356 or the port specified with **-p**). If desired, specify a time-out in seconds; this is the length of time that the server will wait for a connection from the Telnet client after the xfNetLink–xfServerPlus connection has been made. The default is 100 seconds. See example D on page 3-53.

-s

(xfServer only) Run xfServer in secure mode.

-scl*[=level]*

Specify the security compliance level (that is, the available protocols). Valid values are 0, 1, and 2. If not specified, the default is used.

0 Use default protocols for current version of Synergy
1 Use protocols TLS1.0, TLS1.1, TLS1.2 (default)
2 Use protocols TLS1.1, TLS1.2

A value of **0** means that *the available protocols may change with the version of Synergy*; the current default is indicated above. Setting **-scl=2** requires that **-cipher** be set to HIGH. Used with **-cert**, **-cipher**, and **-encrypt**. See "Understanding cipher suites and protocols" on page 3-8 for more information.

-t*time*

(xfServer only) Specify the time interval for monitoring. Used with **-m**.

-text *"description"*

Add the string specified in *description* to the end of the command line when **rsynd** is started. If *description* contains %s, it is replaced with the IP address of the xfNetLink or xfServer client. (Other *%letter* variables should not be used.) Enclose the string in quotation marks but do not include quotation marks *within* the string. This description displays in the list of currently running processes (e.g., **ps -ef** command) and enables you to distinguish among **rsynd** processes when several are running.

-u*persona*

xfServer: Specify *persona* as the default user name or user name and password when running xfServer in non-secure mode. If you include the password, separate the user name and password with a forward slash. The password must be encoded. This option is required when **-encrypt** is used in non-secure mode. Used with **-n**. See "Security modes" on page 3-31 for more information.

xfServerPlus: Specify *persona* as the user name or user name and password under which to run children launched by xfServerPlus. If you include the password, separate the user name and password with a forward slash. The password must be encoded. If **-u** is not specified, the account of the user starting **rsynd** is used. Used with **-w**.

-v

Display the **rsynd** version.

-v6

Server has an IPv6 address. Default is IPv4. If the server's OS supports IPv4 and IPv6 connections on the same port, specifying **-v6** means both IPv4 and IPv6 are supported. If the server's OS *does not* support IPv4 and IPv6 connections on the same port, specifying **-v6** results in *only* IPv6 support.

-w

(xfServerPlus only) Enable xfServerPlus. See the *xfNetLink & xfServerPlus User's Guide* for more information on xfServerPlus.

UNIX usage

You can place a space between an option and its argument if desired.

A. The following example shows how to start xfServer on port 2445 in secure mode:

```
rsynd -p 2445 -s
```

B. The following example shows how to start xfServer in non-secure mode with a default user account on the default port (2330). We ran **setruser** to get the encoded password string. Note that the username/password string should be enclosed in quotation marks.

```
rsynd -n -u "Tiger/\331\247B\200\320\247A\343\027\270R"
```

C. The following example shows how to start xfServerPlus. Because we did not specify a port, xfServerPlus will run on the default port (2356). This command launches the **setruser** utility, which prompts for a user name and password. After you enter the user name and password, the command starts **rsynd** with the specified user name and the encoded password. (Note that **setruser** must be enclosed between grave accent characters (`) as shown.) This technique can also be used to start xfServer in non-secure mode with a default user account.

```
rsynd -w -u `setruser`
```

D. The following example shows how to start xfServerPlus with remote debugging via Telnet enabled. We have specified 2445 as the debug port number (the port the server will listen on for the Telnet client) and 60 seconds as the time-out value. Note that no spaces are allowed between the port number, the colon, and the time-out value. xfServerPlus will use the default port (2356). For more information about debugging via Telnet, see "Debugging Your Remote Synergy Routines" in the "Configuring and Running xfServerPlus" chapter of the *xfNetLink & xfServerPlus User's Guide*.

```
rsynd -rd 2445:60 -w -u `setruser`
```

E. The following example shows how to start xfServer in secure mode on the default port and enable slave encryption using the default cipher, **.pem** filename, and security compliance level.

```
rsynd -s -encrypt
```

F. The following example shows how to start xfServer in secure mode on the default port and enable master encryption and specify values for the other encryption options.

```
rsynd -s -encrypt=master -cipher=high -cert=CERT:myCertFile.pem -scl=2
```

G. Note that if you want slave encryption, and you have used either the **-cert** or the **-cipher** option, you are not required to include the **-encrypt** option. For example, the following command will start xfServer and enable slave encryption using the default **.pem** filename.

```
rsynd -s -cipher=low
```

H. The following example shows how to stop xfServer running on port 2445:

```
rsynd -p 2445 -q
```

OpenVMS options

/ALLOW_PRIVILEGED

Specify that user names that possess system-level privileges may be used as the default user account for xfServer or as the user name under which xfServerPlus server processes run. We do not recommend that you use elevated privileges in a production environment.

/CERTIFICATE=*filename*

Specify the digital certificate file (SSL **.pem** file) to be used for data packet encryption. The file may not require a pass phrase. The default filename and location is **DBLDIR:rsynd.pem**, which will be used if /CERTIFICATE is not specified. *Filename* may use a complete path or a logical. Used with /CIPHER, /ENCRYPT, and /SCL.

/CIPHER=(LOW|MEDIUM|HIGH)

Specify the desired level of encryption. If /CIPHER is not specified, MEDIUM will be used. These values map to specific cipher suite names and protocols, which are determined by the version of OpenSSL being used. If MEDIUM or LOW are specified, /SCL must be set to **1**. You can use the SLC option to the GETFA routine to see which cipher suite and protocol is being used. See GETFA in the "System-Supplied Subroutines and Functions" chapter of the *Synergy DBL Language Reference Manual*. Used with /CERTIFICATE, /ENCRYPT, and /SCL.

/CLIENT_LOGFILE

(xfServer only) Create a log file for each client in the form **DBLDIR:*node*_SRVR_*pid*.log**. This file logs errors as well as startup and shutdown information. If not specified (or if /NOCLIENT_LOGFILE is specified), a log is created only when there is an error.

/DEFAULT_USER=*user_name*

Specify the default user account for anonymous access to xfServer when it is running in non-secure mode. Used with /NOSECURE. *User_name* must be a valid account on the local machine. This option is required when /ENCRYPT is used in non-secure mode. (See "Security modes" on page 3-40 for more information.)

/ENCRYPT=(MASTER|SLAVE|OFF)

Enable data packet encryption for xfServer or—if /XFPL_ENABLE is specified—for xfServerPlus. Specifying OFF is the same as not specifying the /ENCRYPT option. Used with /CERTIFICATE, /CIPHER and /SCL. See "Using Client/Server Encryption" on page 3-7 for details on xfServer encryption or "Using Encryption" in the "Configuring and Running xfServerPlus" chapter of the *xfNetLink & xfServerPlus User's Guide* for details on xfServerPlus encryption.

/FREE_POOL=*nnn*

Specify the number of xfServer file servers to be maintained as a free pool of servers to allow immediate connection of clients. Default value is 2; minimum value is 1.

/INACTIVE_LIMIT=*"nn:nn:nn"*

Specify how long an xfServer process is allowed to persist in an unused state while the free pool is above the free pool limit. Put the value in double quotation marks. The default is 30 minutes ("0:30:00").

/LOG_LEVEL=(ENABLE|FULL|NONE|OPERATOR)

Set the logging level and indicate whether messages should be sent to the console. ENABLE, which logs errors and some status messages, is the program default and is in effect even when /LOG_LEVEL is not specified. FULL logs start and stop messages in addition to errors and status messages. NONE indicates no logging will take place. For FULL or ENABLE, specify a log file with /OUTPUT. If OPERATOR is also specified, messages are sent to the console in addition to a log file. See "Logging" on page 3-41 for additional information.

/NOCLIENT_LOGFILE

(xfServer only) Log only errors for clients. The log file name will be **DBLDIR:*node*_SRVR_*pid*.log**. Specifying this option is the same as simply not specifying /CLIENT_LOGFILE.

/NOSECURE

(xfServer only) Specify that the server should run in non-secure mode. This is the default. See also /SECURE.

/NOUSE_SYNRC

Specify that the xfServer file servers not use the file **SYNRC.COM** in the default directory of the client's user name to set up process-local logical names. See also /USE_SYNRC.

/OUTPUT=*log_file*

Specify the filename to which messages are logged. If this qualifier is omitted, a log file is not produced. By default, messages are logged to the file ***node*_rsynd_*port*.log** in DBLDIR. See "Logging" on page 3-41 for additional information.

/PORT=*nnnn*

Specify the IP port on which xfServer or xfServerPlus listens. The default port is 2330 for xfServer and 2356 for xfServerPlus.

/SCL=*level*

Specify the security compliance level (that is, the available protocols). Valid values are 0, 1, and 2. If not specified, the default is used.

0 Use default protocols for current version of Synergy
1 Use protocols TLS1.0, TLS1.1, TLS1.2 (default)
2 Use protocols TLS1.1, TLS1.2

A value of **0** means that *the available protocols may change with the version of Synergy*; the current default is indicated above. Setting /SCL=2 **r**equires that /CIPHER be set to HIGH. Used with /CERTIFICATE, /CIPHER, and /ENCRYPT. See "Understanding cipher suites and protocols" on page 3-8 for more information.

/SECURE

(xfServer only) Specify that the server requires an encrypted password to be passed along with the user name of the client requesting service from xfServer. The default is /NOSECURE.

/SHUTDOWN*[***=ALL***]*

Stop xfServer or xfServerPlus without killing existing connections. For xfServer, if ALL is specified, existing connections are stopped as well. For more information, see "Stopping xfServer" on page 3-43.

/USE_SYNRC

Specify that the xfServer file servers use the file **SYNRC.COM** in the default directory of the client's user name to set up process-local logical names. See also /NOUSE_SYNRC.

/XFPL_ENABLE=*user_name*

Enable xfServerPlus and run xfServerPlus server processes under the specified user name. See the *xfNetLink & xfServerPlus User's Guide* for more information on xfServerPlus.

/XFPL_FREE_POOL=*nnn*

Specify the number of xfServerPlus logic servers to be maintained as a free pool of processes to allow immediate connection of a client. Default value is 2; minimum value is 1.

The following seven options are the same as those used by the OpenVMS RUN command. Because **rsynd** is a detached process, the defaults for these options would normally come from the PQL_D* **sysgen** parameters; however, we override those defaults during installation. Any options not specifically mentioned below still use the PQL_D* **sysgen** parameters defaults. Note that the values for some of these options are set by the installation in the SYNERGY_SERVER section of **SYNERGY_STARTUP.COM**, but have underlying defaults. See below for details.

/BUFFER_LIMIT=*nnn*

Specify the maximum amount of memory in bytes that an xfServer process can use for buffered I/O operations. The installation sets this option to 65000; if not set, the default is 262144 on Itanium and 65536 on Alpha.

/ENQUEUE_LIMIT=*nnn*

Specify the maximum number of locks that an xfServer process can have outstanding at any one time. The default is 800.

/EXTENT=*nnn*

Specify the maximum to which an xfServer process may increase its physical memory. The installation sets this option to 6000; if not set, the default is 40000 on Itanium and 10000 on Alpha.

/FILE_LIMIT=*nnn*

Specify the maximum number of files that an xfServer process can have open at any one time. The default is 256.

/MAXIMUM_WORKING_SET=*nnn*

Specify the maximum to which an xfServer process may increase its working set size. The installation sets this option to 4000; if not set, the default is 32000 on Itanium and 4000 on Alpha.

/PAGE_FILE=*nnn*

Specify the maximum number of pages of virtual memory that an xfServer process can allocate. The default is 262136 on Itanium and 65536 on Alpha.

/PRIORITY=*nnn*

Specify the priority level at which an xfServer process runs. The default is 5.

The synxfpng Utility

The **synxfpng** utility can be used for troubleshooting network connections to Synergy TCP/IP services including xfServer (**rsynd**), License Manager server (**synd**), and SQL OpenNet (**vtxnetd**). To verify that a client can access one of these services, run **synxfpng** from the client. **Synxfpng** uses *only* TCP/IP API calls and tests *only* whether the client can create and bind a socket; it does not send packets to the server. **Synxfpng** is installed in the dbl\bin directory.

Synxfpng prints a report to the screen, showing each port tested. If there are errors, try running it again with the **-v** option to get more information, including the TCP/IP error number. This information can be used by your network administrator to resolve TCP/IP network socket communication problems.

Syntax

`synxfpng` *[-option] [...] server_name*

Arguments

option

One or more of the following:

-d *size* Specify a packet size for sending to xfServer.

-h Display usage information.

-i*num* Ping the specified (or default) port *num* times. Not valid with **-x**. The default is 5.

-p*port* Ping the specified port.

-r*num* Ping a range of *num* ports, starting with the port specified with the **-p** option or, if **-p** is not specified, with the default port.

-s Ping License Manager server connection. Can be used with **-i**, **-p**, **-r**, **-v**, and **-vers.** (Windows only)

-v Verbose output. See example D on page 3-60 for xfServer sample output.

-v4 Server uses IPv4 addressing. (Default)

-v6 Server uses IPv6 addressing.

-vers Returns the version of the server if it is version 10 or higher. Can be used when pinging xfServer and License Manager server.

-x Ping SQL OpenNet server connection. Can be used with **-p**, **-r**, and **-v**.

server_name

The host name or IP address of the system to ping.

Discussion

You can use **synxfpng** to determine if a DNS lookup problem is the cause of a slow connection. To do this, first run **synxfpng** with the host name as the *server_name* argument, and then run it with the IP address as the *server_name* argument. If the connection is faster using the IP address, there is a DNS lookup problem. Consult your network administrator for assistance in resolving the problem.

By default, **synxfpng** first looks up the IP as an IPv4 formatted address; then, if that fails, it tries IPv6. You can specify that only one or the other be used with the **-v4** and **-v6** options. If both **-v4** and **-v6** are included on the command line, the last one takes precedence.

xfServer

By default, **synxfpng** pings the default xfServer port, 2330. If no service is found on that port, it will increment the port number by 1 (i.e., 2331) and test again. **Synxfpng** will continue incrementing the port number and testing ports until it finds a service or has tested 10 ports. You can specify a non-default port with **-p**, in which case, only that port is tested.

If the SCSPORT environment variable is set in the environment (and you do not specify **-p**), **synxfpng** will ping the SCSPORT port. If no service is found on that port, **synxfpng** will test up to 10 ports as described above. (On Windows, when SCSPORT is set in the registry, it is not read by **synxfpng**.)

By default, **synxfpng** pings 5 times when it finds a service. You can specify a different number of pings with the **-i***num* option.

You can specify the number of ports to test with **-r***num*. By default **synxfpng** will test up to 10 ports, and it stops as soon as it finds the service. But when **-r** is used, **synxfpng** continues pinging until *num* ports are tested, even if it finds a service early in the sequence.

Normally, the packet sent by **synxfpng** is only a few bytes. To specify that a packet of a particular size be sent, specify the size with the **-d** option. The packet will be the size you specify plus the usual packet size. (For example, if it’s usually 28 bytes and you specify -d1024, the packet will be 1052 bytes.)

License Manager server (Windows only)

When used with the **-s** option, by default **synxfpng** pings port 2380, the default License Manager server (**synd**) port. If no service is found on that port, it will increment the port number by 1 (i.e., 2381) and test again. **Synxfpng** will continue incrementing the port number and testing ports until it finds a service or has tested 10 ports. When it finds a service, it pings 5 times.

To ping a non-default **synd** port, use **synxfpng** with the **-s** and **-p** options. The **-s** option also supports specifying a non-default number of pings (**-i**) and pinging a range of ports (**-r**).

SQL OpenNet server

When used with the **-x** option, by default **synxfpng** pings port 1958, the default SQL OpenNet server port. If no service is found on that port, it will increment the port number by 1 (i.e., 1959) and test again. **Synxfpng** will continue incrementing the port number and testing ports until it finds a service or has tested 10 ports.

When used with the **-x** and **-v** options, **synxfpng** provides more verbose SQL OpenNet error reporting than is available with **vtxping**.

To ping a non-default SQL OpenNet port, use **synxfpng** with the **-x** and **-p** options. The **-x** option also supports pinging a range of ports (**-r**).

Usage

A. To ping the server named Tiger on the default xfServer port (2330):

```
synxfpng tiger
```

Sample output:

```
Pinging Synergy/DE xfServer on host [tiger] port 2330

Round trip (28 bytes) 31.933 ms
Round trip (28 bytes) 2.055 ms
Round trip (28 bytes) 0.795 ms
Round trip (28 bytes) 0.951 ms
Round trip (28 bytes) 0.551 ms

Average round trip 7.257 ms
```

B. To ping License Manager server on port 2445, located on the server named Tiger, three times:

```
synxfpng -s -p2445 -i3 tiger
```

C. To ping SQL OpenNet on ports 1955 through 1959 on the server named Tiger:

```
synxfpng -x -p1955 -r5 tiger
```

D. To ping the server named Tiger on the default xfServer port and receive verbose output:

```
synxfpng -v tiger
```

Sample output with the **-v** option (Windows):

```
WinSock status information:
    Description: WinSock 2.0
    SystemStatus: Running
    Using version 1.1
    Supports up to version 2.2

Starting DNS Lookup tiger Wed Oct 03 11:28:44 2012
 -- Complete 127.0.0.1 Wed Oct 03 11:28:44 2012
```

```
Pinging Synergy/DE xfServer on host [tiger] port 2330

Starting socket(AF_INET, SOCK_STREAM, IPPROTO_TCP) -- Completed
Starting bind() -- Wed Oct 03 11:28:44 2012

Starting connect() -- Wed Oct 03 11:28:44 2012
     -- Completed Wed Oct 03 11:28:44 2012

Starting setsockopt(, SOL_SOCKET, SO_KEEPALIVE,,) -- Completed
Starting setsockopt(, IPPROTO_TCP, TCP_NODELAY,,) -- Completed

Starting Ping Wed Oct 03 11:28:44 2012

Round trip (28 bytes) [v10.0.3 server] 3.464 ms
Round trip (28 bytes) [v10.0.3 server] 8.340 ms
Round trip (28 bytes) [v10.0.3 server] 0.772 ms
Round trip (28 bytes) [v10.0.3 server] 28.167 ms
Round trip (28 bytes) [v10.0.3 server] 0.598 ms

Average round trip 8.268 ms
Socket closed Wed Oct 03 11:28:49 2012
```

See also

SCSPORT in the “Environment Variables” chapter of *Environment Variables & System Options*.

The synckusr Utility (Windows)

The **synckusr** utility can be used to diagnose xfServer problems that are related to user account setup on Windows.

Syntax

```
synckusr [-h] [-p] [-r] [-s] [user_name] [service_name]
```

Arguments

-h

Display usage information.

-p

Authenticate the password for the specified *user_name*.

-r

Display the user name and password (in masked form) currently being used by the runtime client and attempt to authenticate the user name and password.

-s

Display the default file path for the specified *service_name*. Used with **-p**.

user_name

The user name to look up.

service_name

The name of the xfServer service for which you want to look up the default file path. Used with **-s**.

Discussion

If there is an xfServer client connection problem, you may get this message:

Cannot log on user: <user_name>
Logon failure: unknown user name or bad password

To troubleshoot this problem, first run **synckusr** with the **-r** option on the client machine. This will tell you which user name and password the client is sending to xfServer. Then, run **synckusr** on the server machine with the **-p** option. This will tell you which user name and password xfServer is attempting to authenticate. You may find that there is more than one account with the same user name across domains. If so, you may need to disable one of the accounts, change the user name specification (in **setruser**) to include the domain or machine name, or create a different user name to resolve the conflict. If multiple accounts are not the problem, verify that you have set the user name and password correctly with **setruser**.

If files without a path specification are not going to the desired (or expected) location on the server, run **synckusr** on the server with the **-s** and **-p** options. This will display the default file directory for the specified xfServer service and user name. You can run the Synergy Configuration Program to determine the service name (the default service is **SynSrv**). To change the default directory, set the RSFILPATH environment variable. See RSFILPATH in the “Environment Variables” chapter of *Environment Variables & System Options*.

Usage

To display the user name and password

To display the user name and password that will be passed through the client connection to xfServer, run **synckusr** with the **-r** option on the client machine. Do not specify a user name; the **-r** option obtains the user name from the client machine. This option also performs a lookup on the user account and attempts to authenticate it. This option returns the user name and password, along with the domain in which the account was found, the Windows security ID (SID), and, if the user is a member of the administrators group, that will display as well.

For example, on the client machine you’d enter

```
synckusr -r
```

Sample output when RUSER is set to a user name and password with **setruser**:

```
RUSER <mark> from HKEY_CURRENT_USER\Software\Synergex
Password is <\330T\301\313\312S\330I\305U\316\032l\343>
Computer Name SPIKE
Encoded User mark
Non Domain specified user pperkins Checking if part of local
administrator's group(s)
Localized Admin user: Administrator Local Admin group: Administrators
User is part of Administrator Group
User account is found in domain: ABC_CORP
User SID is: S-1-5-21-516626026-640904803-9522986-1071
Account has been authenticated
```

Sample output when RUSER is set to SSPI with **setruser**:

```
RUSER <SSPI> from HKEY_CURRENT_USER\Software\Synergex (Windows
authentication)
No Password
Windows authentication will be used when connecting to a Windows server
```

Sample output when RUSER is set in the environment:

```
<RUSER=mark> from environment
No password, account lookup and authentication not possible
User account is found in domain: ABC CORP
User SID is: S-1-5-21-1379799261-1886764798-3495251409-1195
Account has not been authenticated
```

Sample output when RUSER is not set, but the user name is found on the local machine:

```
RUSER not set. <mark> from GetUserName system call (implied RUSER)
No Password, account lookup and authentication not possible
User account is found in local computer: SPIKE
User SID is: S-1-5-21-487314201-320540311-1472111-1003
Account has not been authenticated
```

To look up and authenticate a user account

To perform a user account lookup and authenticate it, run **synckusr** on the xfServer machine with the **-p** option and specify the user name you want to look up. You'll be prompted to enter the password. **Synckusr** will return the domain in which the account was found and the Windows security ID. If the user is a member of the administrators group, that will display as well. Running **synckusr** with the user name and password enables you to distinguish between multiple accounts with the same user name, but different passwords. Use this option if you are running xfServer in secure mode.

For example, on the xfServer machine you'd enter

```
synckusr -p Mark
```

Sample output:

```
User account is found in domain: ABC_CORP
User SID is: S-1-5-21-516626026-640904803-9522986-1071
```

To look up and display the domain for a user account

To perform a user account lookup and display where the user name was found, run **synckusr** on the xfServer machine and specify the user name you want to look up. This is most useful if you are running in non-secure mode; for secure mode, use the **-p** option. **Synckusr** will return the domain in which the account was found and the Windows security ID (SID). This will tell you which client user name xfServer will attempt to authenticate (log in); this can be useful if users have multiple accounts. (To actually authenticate the user name, use the **-p** option.)

For example, on the xfServer machine you'd enter

```
synckusr Mark
```

Sample output:

```
User account is found in domain: ABC_CORP
User SID is: S-1-5-21-516626026-640904803-9522986-1071
```

To display the default file directory

To display the default file directory, run **synckusr** with the **-p** and **-s** options on the xfServer machine and specify the user name and service name. This will tell you the directory that files from that user without a path specification will be placed in. If the default file directory is set with RSFILPATH, it will also tell you where RSFILPATH is set (that is, server-level, user-lever, etc.).

For example, on the xfServer machine you'd enter

```
synckusr -p Mark -s SynSrv
```

Sample output when RSFILPATH is set for a specific instance of xfServer:

```
Password: *********
Default directory: <c:\Synergy\Data> from RSFILPATH in server Synrc
(HKLM)
User account is found in local computer: SPIKE
User SID is: S-1-5-21-345376971-725947319-313073093-1007
Account has been authenticated
```

Sample output when RSFILPATH is not set and the default directory is used. This example shows the default directory on Vista.

```
Password: ********
Default directory: <C:\Users\Public\Documents>
User account is found in domain: ABC_CORP
User SID is: S-1-5-21-516626026-640904803-9522986-1145
Account has been authenticated
```

If the user specified with **-p** has never logged on to the server machine, you will see "Key missing, account setup error…The parameter is incorrect." However, the information about the default directory will then display correctly.

See also

- RUSER in the "Environment Variables" chapter of *Environment Variables & System Options*.
- RSFILPATH in the "Environment Variables" chapter of *Environment Variables & System Options*.
- "The setruser Utility" on page 3-66.

The setruser Utility

This program is used to generate the encoded string that is used to set the RUSER environment variable or registry setting on the client machine.

Syntax

`setruser` *[-option]*

Arguments

option

One of the following:

WIN

<none> Create or update the RUSER environment variable in the registry for the current user (local setting).

-d Delete the RUSER setting from the registry.

-g Create or update the RUSER environment variable in the registry for all users (global setting).

-h Display usage information.

-n Display the encoded string to the screen without updating the registry.

UNIX

<none> Display the encoded string to the screen.

Discussion

When you run **setruser** without any options, it will prompt you for the user name and password that you have set up on the server for secure access.

- On Windows, it returns the encoded string and uses this string to set RUSER in the registry under **HKEY_CURRENT_USER\ Software\Synergex**. This sets RUSER for the user who is currently logged in. If RUSER is set at the global level on this machine, a message to that effect will display to inform you that the local (user-specific) setting will be used.
- On UNIX, it returns the encoded string, which you can then use to set RUSER.

WIN

When you run **setruser** with the **-g** option, it will prompt you for the user name and password that you have set up on the server for secure access, and then return the encoded string and use that string to set RUSER in the registry under **HKEY_LOCAL_MACHINE\ Software\Synergex**. (If 32-bit is installed on a 64-bit machine, you may also see it set in **HKEY_LOCAL_MACHINE\ Software\Wow6432Node\Synergex**.) This sets RUSER globally for all users on the machine. If there is a user-specific RUSER setting on the machine, it is not altered, and you will see a warning message that a local RUSER setting exists on the machine and will be used.

The runtime reads RUSER first from the environment. If RUSER is found there, it is used. If RUSER is not set in the environment, the runtime checks **HKEY_CURRENT_USER** for the logged-in user and uses the setting found there. If RUSER is not set at the user level, then the runtime checks for a global setting in **HKEY_LOCAL_MACHINE**.

When RUSER is set in the **Synrc** node in the registry, as is required when configuring xfServerPlus for remote data access, it is read into the environment when **rsynd** starts up, and it is therefore considered to be set in the environment.

The user name "SSPI" (all uppercase) is reserved for use with Windows authentication. Setting RUSER to SSPI and *not* specifying a password (just press RETURN at the password prompt) indicates to xfServer that Windows authentication should be used. Setting RUSER to SSPI is necessary only if there is a global RUSER setting that you want to override. (See "Understanding xfServer security" on page 3-16 for detailed information about security.) If you set RUSER to SSPI and specify a password, or if you set RUSER to sspi (lowercase), it will be treated as an ordinary RUSER setting, but we do not recommend doing this, as it can lead to confusion.

To delete a local RUSER setting from the registry, run **setruser** with the **-d** option. To delete a global RUSER setting, run **setruser** with the **-d** and **-g** options. If there is another RUSER setting on the machine (e.g., you are deleting a local setting and a global setting still exists), a message to that effect will display.

The **-n** option prompts you for the user name and password, and then displays the encoded string to the screen without updating the registry. You can use this string to set RUSER in the environment.

If you have an account with the same user name on both the local machine (i.e., the client) and on a Windows domain, or on multiple domains, and you want to use a specific domain account, when prompted for the user name type *user_name@domain_name* or *domain_name\user_name*. (This applies when running **setruser** without any options and when running it with the **-n** option or the **-g** option.) If you don't specify the domain, xfServer will find the account on the local machine first, and then display an error because the password doesn't match.

UNIX

On UNIX, when you run **setruser**, you’ll be prompted to enter the user name and password. When the encoded string displays, use it to set RUSER:

```
RUSER=setruser_value ;export RUSER
```

Or, you can automate this process by executing this shell command. This command invokes **setruser** (which prompts you for the user name and password), assigns the output to RUSER, and exports it. **Setruser** must be enclosed between grave accent characters (`) as shown below or the command will not execute.

```
RUSER=`setruser` ;export RUSER
```

See also

RUSER in the “Environment Variables” chapter of *Environment Variables & System Options*.

Troubleshooting xfServer

General troubleshooting tips

When you are experiencing problems with xfServer, the first thing to do is to check the log. Even when the error is displayed on the client, the server-side logs may contain useful troubleshooting information. On all systems, some basic logging is done by default, with additional logging available.

- **Windows:** By default, **rsynd** logs its version, start/stop information, and critical errors to the Windows event log. You can use the "Enable logging" and "Verbose logging" options in the Synergy Configuration Program to turn on additional logging. (See page 3-13.) The Monitor utility may also be of some assistance, especially when you require information regarding files that are opened, locked, etc. (See "The Monitor Utility for Windows" in the "General Utilities" chapter of *Synergy Tools*.)
- **UNIX:** By default, **rsynd** logs start/stop information and critical errors. (See "Logging" on page 3-33.) The Monitor utility may also be of some assistance when troubleshooting. (See "The Monitor Utility for UNIX" in the "General Utilities" chapter of *Synergy Tools*.)
- **OpenVMS:** By default, **rsynd** logs errors and some status messages. You can use the /LOG_LEVEL=FULL qualifier for additional logging. (See "Logging" on page 3-41.)

Start-up errors (Windows)

When I start the xfServer service, I get the error "A service-specific error occurred: 99."

See the Windows Event Viewer for additional information about this error.

Login errors

What does it mean when a client receives an "Unknown host '*nnn*' in server spec" error?

It means that the client has no TCP/IP connection to the specified host. Use **synxfpng** to determine if the client can connect to the server machine via TCP/IP. See "The synxfpng Utility" on page 3-58 for more information.

What should I do if my client is receiving a "Bad user name, login rejected on *servername*" error?

This error can occur when you are running in secure mode. There are several problems which can cause this error message.

- RUSER is not set on the client.
- RUSER is set on the client, but the user name and password don't match an account on the server.
- RUSER is set on the client, but the user account has not logged in to the server console. Some systems may require the user to *physically* log in to the server prior to making a connection to xfServer.

- You are using Windows authentication security, and the client is not in the same domain as the server or did not log in successfully to the domain.

If you are using RUSER security, you should first verify that RUSER is set on the client and that the client has an account on the server using the same name and password. On Windows, you can run **synckusr** on the server to see where xfServer is finding the client user name. You can also run **synckusr** on the client to see what user name and password are being sent to xfServer. See "The synckusr Utility (Windows)" on page 3-62 for details. Refer to the following sections for information on how and where RUSER is set:

- "The setruser Utility" on page 3-66
- RUSER in the "Environment Variables" chapter of *Environment Variables & System Options*
- For Windows, "Understanding xfServer security" on page 3-16
- For UNIX, "Security modes" on page 3-31
- For OpenVMS, "Security modes" on page 3-40

File permissions (UNIX)

How do I get xfServer to create files with user-defined permissions?

To ensure that the desired permissions are used, set the UMASK environment variable. To set UMASK for a specific user, add the following line to the user's **.synrc** file:

```
UMASK=mask
```

where *mask* is the octal umask value you want to set. Refer to the octal form of umask(1) in the UNIX man pages for details. The next client connection for that user will use umask. To affect all users, you can set UMASK in the **/etc/synrc** file and then restart **rsynd**. You can also set UMASK in the environment before starting **rsynd** to affect all users.

Some security implementations cause the umask value to be cleared. You must use the UMASK environment variable in those instances.

File access (Windows)

What should I do when I get a "file not found" message when starting an xfServer service?

Remove the xfServer service with the **rsynd -x** command, and then re-register it with **rsynd -r**. You can also remove and re-register rsynd from the Synergy Configuration Program; see "Stopping xfServer" on page 3-28 and "Starting xfServer" on page 3-27.

Why do I get a protection error accessing files on xfServer, even though I can access files using mapped drives.

The default search path for user authorization on mapped drives is different from the system call we make to authorize users. The system call LookupAccountName is used to look up security information for a user name. It first checks well-known system IDs, then checks local accounts with administrator privileges, and then checks the primary domain. If the user name is still not found, it checks trusted domains.

If you want to know where the user account is being found by xfServer, run the **synckusr** utility on your xfServer machine. See "The synckusr Utility (Windows)" on page 3-62 for details. Once you know which machine provided the log-in information, you can check the access permissions for that account on the machine that is authorizing the user. We recommend that each user account have a unique name in a network where several machines could authorize the account.

xfServer is running, but I cannot connect to it from my client.

This is often a DNS problem or a TCP/IP configuration problem. Try using the **synxfpng** utility on the client to assist your network engineer in troubleshooting the problem. See "The synxfpng Utility" on page 3-58 for more information.

Rsynd: shutting down and timing out

Connections to the xfServer daemon, **rsynd**, are normally closed down by the client through normal exit processing, thereby freeing licenses, closing files, and releasing locks. However, if the client dies with a network-related error or a segmentation fault, or if the client machine is turned off or the network fails, **rsynd** never gets the shut-down request. In such cases, the continued operation of the server process is determined under the rules of the TCP/IP protocol and a timer called the keepalive timer.

The TCP keepalive timer, which is different on all systems, sends a request on idle sockets in an attempt to connect to the client machine and determine if the connection is still valid. (A socket is considered idle if there have been no calls from the client on that socket for the length of time specified in the keepalive timer.) If the connection is broken, **rsynd** can determine this and shut down, freeing licenses, releasing locks, and closing files.

Because the wait time for the keepalive timer is normally about two hours on most Windows and UNIX systems, we set it to ten minutes each time a socket is opened. This setting temporarily overrides the system keepalive setting.

On Windows, the connection recovery feature, which is enabled by default, has its own keepalive timer setting (see "KeepAlive" on page 3-21). The ten minute setting described here applies only when connection recovery is disabled.

On OpenVMS, the amount of wait time is much less; you can check the setting and reset it if necessary in TCPIP. See your OpenVMS operating system documentation or supplier for details.

4

Configuring Connectivity Series

Information about troubleshooting Connectivity Series components, as well as details on logging and error messages, can be found in the product manuals: *SQL Connection Reference Manual* and *xfODBC User's Guide*.

What Is Connectivity Series?

Synergy/DE Connectivity Series consists of SQL Connection, xfODBC, and SQL OpenNet.

SQL Connection is an API that enables you to access third party databases, such as Oracle, from a Synergy application. xfODBC is a group of utilities that enables you to access Synergy data from third party ODBC-enabled applications, such as Microsoft Access.

Both SQL Connection and xfODBC can be used stand-alone or in a client/server configuration. When used in a client/server configuration, the SQL OpenNet server provides the network layer. The client can be installed as either Connectivity Series or xfODBC Client, which is a thinner client that does not require the installation of Core Components. (For details on the xfODBC Client installation, see "xfODBC requirements and installation" in the "Welcome to xfODBC" chapter of the *xfODBC User's Guide*.)

This chapter describes how to configure SQL OpenNet for use with SQL Connection and xfODBC, as well as how to configure SQL Connection for stand-alone or client use and xfODBC for client use (Windows only). For information about using SQL Connection and xfODBC, see the product manuals: *SQL Connection Reference Manual* and *xfODBC User's Guide*.

Using Data Packet Encryption for SQL OpenNet

The SQL OpenNet data packet encryption feature enables you to encrypt the transfer of sensitive data across a network. SQL OpenNet interfaces with a third-party library, OpenSSL, to provide SSL support for secure data transport between client and server.

To use SSL encryption, the server and all clients must be version 10.3.3 or higher.

Using encryption can affect performance because data must be encrypted and decrypted on both sides of the SQL OpenNet connection. The cipher negotiation between client and server, even though it happens only when a connection is established, also takes time.

To implement encryption, you must start **vtxnetd** or **vtxnet2** with the **-e** option. You specify the SSL certificate file, the private key file, and optionally a list of accepted protocols (TLS versions). You can specify one or more of the following protocols: TLS 1.0, 1.1, 1.2. By default, all three are accepted. Bear in mind that because protocol support varies by operating system, older systems may not support newer protocols. For example, Windows Vista, Windows Server 2008, and HP-UX (PA-RISC) do not support any protocols above TLS 1.0. Additionally, as new threats and vulnerabilities are found, Synergex may need to update the default for accepted protocols to maintain a high level of security.

Follow these steps to set up SQL OpenNet to use SSL encryption. For details on which version of OpenSSL is required for your operating system, see "OpenSSL Requirements" on page 1-11. For additional information on OpenSSL, see Synergex KnowledgeBase article 100001979.

1. Install OpenSSL on your server machine.

    - For Windows, go to www.openssl.org/community/binaries.html and follow the link to download the version for your platform. See Synergex KnowledgeBase article 100001979 for details on file placement.
    - For Unix, use the OpenSSL libraries installed with your operating system or download the OpenSSL security kit specific to your platform and version from your operating system vendor.
    - For OpenVMS, use the operating system-supplied libraries, HP SSL, available from h71000.www7.hp.com/openvms/products/ssl/ssl_download.html.

2. Ensure that the OpenSSL shared libraries are in the correct location on the server and have been added to the correct path. The library path must be set before starting **vtxnetd** or **vtxnet2**.

    - On Windows, the libraries must be located on the SQL OpenNet machine in the dbl\bin and connect directories that correspond to the bitness of your machine (as explained in Synergex KnowledgeBase article 00001979).
    - On UNIX, the OpenSSL libraries are installed in a standard location determined by the operating system. You don't need to move them or set the path.

- On OpenVMS, during installation SYSSLLIB is set in SYS$MANAGER:SYNERGY_STARTUP.COM to the full path and filename of **SYNSSLLIB.EXE**, the SSL runtime support file.

3. Create a certificate file and a private key file. You may name these anything you like and put them anywhere on the server. Note that the certificate file cannot include a pass phrase. For information on creating certificate and key files, see "Requesting a certificate" and "Testing your HTTPS setup locally" in the "Synergy HTTP Document Transport API" chapter of the *Synergy DBL Language Reference Manual*.

4. Install and configure OpenSSL on the client machines.

   - On Windows clients, copy the OpenSSL libraries to both the dbl\bin directory and the connect directory.
   - On UNIX clients, the library path is used to find the OpenSSL libraries. If you used the **setsde** script to set up your Synergy environment, the path will be correctly set. You can run the **dltest** utility (installed in the Synergex\SynergyDE\connect directory) to find your shared library path and determine if Synergy can find the necessary DLLs.

5. Start **vtxnetd** or **vtxnet2** with the **-e** option. Specify the certificate and the private key file, and optionally specify the protocol (TLS levels). See "The vtxnetd and vtxnet2 Programs" on page 4-27 for details.

   To determine whether the SQL OpenNet service is set to use data packet encryption (SSL), start **vtxnetd** or **vtxnet2** with the **log** option, and then use **vtxping** to test the connection. If SSL encryption is in use, the log file will include something like the following:

```
Thu Mar 17 10:38:02 2016 - Version 4.0.0.17.
Thu Mar 17 10:38:02 2016 - Starting vtxnetd (pid: 6128).
Thu Mar 17 10:38:02 2016 - Server port 1987
Thu Mar 17 10:38:02 2016 - SSL enabled
Thu Mar 17 10:38:02 2016 - SSL compile/library: OpenSSL 1.0.2d 9 Jul
2015/OpenSSL 1.0.2d 9 Jul 2015
Thu Mar 17 10:38:02 2016 - Creating 'listen' queue (max: 10)
Thu Mar 17 10:38:02 2016 - Setup done. Going into 'accept' loop
Thu Mar 17 10:38:07 2016 - Starting a TCMHOST thread (parms: 128
VTX11_12)
```

   If the log file includes an "SSL compile/library:…No such file or directory" error, the path to the certificate or private key file is incorrect.

   If the log file includes an "SSL compile/library:…problems getting password" error, the private key file includes a pass phrase, which is not supported. For information on removing a pass phrase, see "Testing your HTTPS setup locally" in the "Synergy HTTP Document Transport API" chapter of the *Synergy DBL Language Reference Manual.*

   An "INVVER: NET version mismatch…" error may mean that a client from a prior version of Connectivity Series is attempting to connect to the SQL OpenNet service while SSL encryption is enabled. To use SSL encryption, the server and all clients must be version 10.3.3 or higher.

SQL Connection and xfODBC on Windows

This section explains how to configure SQL Connection and xfODBC for remote use with SQL OpenNet and for local (stand-alone) use.

- To run **SQL Connection in a client/server configuration**, see "SQL Connection and xfODBC: using SQL OpenNet" on page 4-6 for server set-up and "SQL Connection: configuring client or stand-alone access" on page 4-10 for client set-up.
- To run **SQL Connection stand-alone**, see "SQL Connection: configuring client or stand-alone access" on page 4-10. (Stand-alone is defined as any connection that doesn't use SQL OpenNet.)
- To run **xfODBC in a client/server configuration**, see "SQL Connection and xfODBC: using SQL OpenNet" on page 4-6 for server set-up and "xfODBC: testing the network connection for client access" on page 4-12 for client set-up.
- To run **xfODBC stand-alone**, see "xfODBC: stand-alone access" on page 4-12.

Understanding SQL OpenNet on Windows

On Windows, you can run SQL OpenNet with either the **vtxnetd** program (which is the default) or the **vtxnet2** program. Both are started by the service program **sqld**. The one you should use depends on the data source you need to connect to. For program syntax, see "The vtxnetd and vtxnet2 Programs" on page 4-27.

The vtxnetd program

The **vtxnetd** program is a multi-threaded server, which listens on the selected port and starts threads to service the connection requests and perform the requested work. The driver name (e.g., **vtx12_SQLNATIVE**) in the connect string determines the DLL that **vtxnetd** attaches to its worker thread to service the connection. You can use **vtxnetd** in the following circumstances:

- Oracle and Synergy databases
- SQL Server databases when accessed with the **vtx12_SQLNATIVE** driver
- Certain other databases when accessed with the **vtx11** (ODBC) driver

Using the multi-threaded **vtxnetd** reduces start-up overhead on initial connections and has a small performance advantage over the multiple program approach of **vtxnet2** (see below). If a connection request is made via **vtxnetd** to a driver that does not support multi-threading, the connection is rejected.

The **vtxnetd** program offers the ability to specify how existing connections are handled when SQL OpenNet is shut down, which **vtxnet2** does not support. See "Stopping and removing SQL OpenNet" on page 4-8.

The vtxnet2 program

The **vtxnet2** program is a listener daemon, which listens on the requested port and creates child processes to service client requests. The driver name portion of the connect string (e.g., **!vtx12_SQLNATIVE**) determines the server executable and its associated DLL, which are used to create the process to service the connections. To use **vtxnet2**, you must edit the **opennet.srv** file; see "Customizing the opennet.srv file" on page 4-9.

The sqld program

The service program **sqld** starts either the **vtxnetd** or **vtxnet2** program, and then polls the program periodically to verify that it is still running. The service name for **sqld** is **SynSQL**, and the display name is "Synergy/DE OpenNet Server". **Sqld** reads the **opennet.srv** file, which contains the daemon start-up commands and parameters as well as needed environment variable settings. See "The sqld program" on page 4-13 and "Customizing the opennet.srv file" on page 4-9 for more information.

SQL Connection and xfODBC: using SQL OpenNet

To use SQL OpenNet with SQL Connection or xfODBC, you need to do the following on your SQL OpenNet server machine:

- Specify the SQL OpenNet port number in the TCP/IP services file.
- Start and test the server.

Specifying the port number

You must specify the port number used by SQL OpenNet in the TCP/IP services file. Open the **services** file in %windir%\system32\drivers\etc and add this line:

```
vtxnet nnnn/tcp #Synergy/DE SQL OpenNet server
```

where *nnnn* is the port number. The default port number is 1958, but you can specify any unused port.

⚠ If you use a non-default port, you must also do the following:

- Change the port specified with the **-p** option on the command line that starts **vtxnetd**/**vtxnet2** in the **opennet.srv** file. See "Customizing the opennet.srv file" on page 4-9 and "The vtxnetd and vtxnet2 Programs" on page 4-27.
- Specify the port on the client. For SQL Connection, see page 4-10 (Windows), page 4-17 (UNIX), or page 4-24 (OpenVMS). For xfODBC, see "Setting Up Access with DSNs" in the "Configuring Data Access" chapter of the *xfODBC User's Guide*.

Registering, starting, and testing SQL OpenNet

SynSQL must be started by a user with administrator privileges.

1. Before starting SQL OpenNet, you must first register the **SynSQL** service. Do one of the following:
   - Start the Synergy Configuration Program (from Windows Control Panel, select Synergy Control Panel > Synergy Configuration Program), go to the Connectivity Series tab, and click the Add Service button. (If the Add Service button is disabled, either **sqld.exe** is not installed or the service is already registered. If the latter, the Start Service button will be enabled.)
   - At a command prompt enter

     ```
     sqld -r
     ```

   You can register and start the **SynSQL** service in one step with the **sqld -rs** option.

2. There are several ways to start **SynSQL**. Do one of the following:
   - In the Synergy Configuration Program, go to the Connectivity Series tab and click the Start Service button.
   - Go to Administrative Tools > Component Services, and expand the Services node. Select Synergy/DE OpenNet Server and click the Start button.
   - At a command prompt enter

     ```
     net start synsql
     ```

3. To verify that the server is running, at a command prompt run **vtxping**:

   `vtxping` *[*`-p`*port] server_name*

   where *port* is the port number that the server is running on, and *server_name* is the host name of the SQL OpenNet machine. If no port is specified, the default port specified in the **services** file is used.

   If the server started successfully, you'll receive a "vtxnetd is alive and kicking" message. If there's a problem, you'll see an error message. See "The vtxping Utility" on page 4-32 for more information. You can check the Windows event log for additional information. You may want to start **sqld** with the **-l** option for more detailed logging; see "The sqld program" on page 4-13.

   SQL OpenNet is now configured. For SQL Connection, you must license your database drivers before you can run the example programs. See "Testing SQL Connection (client or stand-alone)" on page 4-10 for information on the example programs. See chapter 2, "Configuring License Manager," for licensing information.

Stopping and removing SQL OpenNet

You must stop the service before you can remove (unregister) it. When you stop SQL OpenNet, no new connections can be made. The behavior of existing connections depends on whether you are using **vtxnetd** or **vtxnet2**:

- **vtxnetd**: By default, existing connections are terminated after a delay of up to 10 seconds, which allows underlying processes time to complete. You can change the delay time by editing the command line in the **opennet.srv** file.

  To change this behavior such that existing connections are *not* terminated, remove the **-w** option from the command line in **opennet.srv**. This is not recommended, as the **vtxnetd** program will not completely shut down until all child processes are terminated. See "Shutting down vtxnetd on Windows (-w)" on page 4-30 for details.

- **vtxnet2:** Existing connections are not stopped. Only the third-party applications that are connected to the Synergy data can stop existing processes. The **vtxnet2** program does not completely shut down until all child processes have terminated.

To *stop* and *remove* SQL OpenNet,

- At a command prompt enter

  ```
  sqld -x
  ```

To *stop* SQL OpenNet, do one of the following:

- In the Synergy Configuration Program (from Windows Control Panel, select Synergy Control Panel > Synergy Configuration Program), go to the Connectivity Series tab and click the Stop Service button.
- Go to Administrative Tools > Component Services, and expand the Services node. Select Synergy/DE OpenNet Server and click the Stop button.
- At a command prompt enter

  ```
  sqld -q
  ```

- At a command prompt enter

  ```
  net stop synsql
  ```

To *remove* SQL OpenNet, do one of the following:

- In the Synergy Configuration Program, go to the Connectivity Series tab and click the Remove Service button.
- At a command prompt enter

  ```
  sqld -x
  ```

Customizing the opennet.srv file

Editing the **opennet.srv** file is optional. The **opennet.srv** file, located in synergyde\connect, is read by **sqld**. It contains the command line that is used to start **vtxnetd** or **vtxnet2**. By editing that command line, you can use a port other than the default, run multiple servers, and so on. You can also define environment variables in the **opennet.srv** file.

> The **opennet.srv** file is not overwritten when you upgrade Connectivity Series, nor is it removed when you uninstall. We distribute a file named **opennet_base.srv** (also located in synergede\connect), which contains default settings and can be used as a reference.

See below for information on using **vtxnet2** instead of the default **vtxnetd**, defining environment variables, caching, and changing the polling interval. For information on the following, see the **vtxnetd** and **vtxnet2** program syntax in "The vtxnetd and vtxnet2 Programs" on page 4-27:

- Using a non-default port
- Server-side logging
- Running multiple SQL OpenNet servers

Using vtxnet2

The default program, **vtxnetd**, is a multi-threaded server that generally performs better than **vtxnet2**. However, you can use **vtxnet2** when you want SQL OpenNet to be a listener daemon that creates child processes rather than a multi-threaded server. The start-up lines for both **vtxnetd** and **vtxnet2** are included in the **opennet.srv** file. To use **vtxnet2**, just comment out the **vtxnetd** line and enable the **vtxnet2** line. See "Understanding SQL OpenNet on Windows" on page 4-5 for additional information about **vtxnetd** and **vtxnet2**.

Defining environment variables

Any environment variables used by SQL OpenNet, such as those that specify the location of your data files, can be defined in the **opennet.srv** file. Define environment variables towards the beginning of the file, before the **vtxnetd** or **vtxnet2** start-up line.

Caching

You can enable system catalog caching for xfODBC by editing the "syngenload" line in the **opennet.srv** file. For more information, see "System Catalog Caching" in the "Configuring Data Access" chapter of the *xfODBC User's Guide*.

Changing the polling interval

By default, the **sqld** program polls (checks) **vtxnetd/vtxnet2** every 10 minutes to verify that it is still running. If **vtxnetd/vtxnet2** stops unexpectedly, users will be unable to connect, and yet no error will be recorded in the Windows event log (nor will the status of the **SynSQL** service change in the Component Services dialog box) until the next poll takes place. Consequently, you may want a shorter polling interval, so that should **vtxnetd/vtxnet2** stop unexpectedly, **sqld** will report the event promptly in the Windows event log.

To change the polling interval, in the **opennet.srv** file, remove the comment (#) at the beginning of the OPENNET_POLL_TIME line and specify the desired polling interval in milliseconds. For example, to set the polling interval to one minute, you'd enter

```
OPENNET_POLL_TIME=60000
```

SQL Connection: configuring client or stand-alone access

This section describes how to configure SQL Connection as a client and how to test your client/server or stand-alone configuration.

Configuring SQL Connection (client)

1. Add the following line to your TCP/IP **services** file in %windir%\system32\drivers\etc:

   ```
   vtxnet nnnn/tcp #Synergy/DE SQL Connection
   ```

   where *nnnn* is the port number. The default port number is 1958, but if you configured SQL OpenNet to run on a different port, specify that port number here.

2. Run **vtxping** to test your connection to the server:

   ```
   vtxping [-pport] server_name
   ```

   where *port* is the port number on which the client will connect to SQL OpenNet, and *server_name* is the host name of the SQL OpenNet machine. If no port is specified, the default port specified in the **services** file is used.

   If the network connection is working properly, you'll receive a "vtxnetd is alive and kicking" message. If there's a problem, you'll see an error message. See "The vtxping Utility" on page 4-32 for more information.

Testing SQL Connection (client or stand-alone)

1. Set the SQL_CONNECT environment variable to specify a connect string.

   - On a client configuration, use this connect string:

     ```
     net:connect_string@[port:]host!driver_name
     ```

     where *connect_string* contains the driver-specific information to be passed to the database driver, *port* is the port number on which SQL OpenNet is running (required for non-default port), *host* is the server system's unique name, and *driver_name* is the driver name (e.g., **vtx12_SQLNATIVE**).

The syntax for *connect_string* depends on the driver. For a list of driver names and the connect syntax to use with them, see "Building connect strings" in the "Creating SQL Connection Programs" chapter of the *SQL Connection Reference Manual*.

For example, to connect to an SQL Server database on a remote Windows machine, you might enter

```
set SQL_CONNECT = net:user_name/manager/mydsn@win_serv!vtx12_SQLNATIVE
```

- On a stand-alone configuration, use this connect string:

  driver:connect_string

  where *driver* is the name (e.g., vtx12_SQLNATIVE) of the database driver, and *connect_string* contains the required information to be passed to the database driver. The syntax for *connect_string* depends on the driver. See "Building connect strings" in the "Creating SQL Connection Programs" chapter of the *SQL Connection Reference Manual* for complete information.

  For example, to connect to an Oracle 11 database, the connect string consists of the user ID and password. So, you might enter

  ```
  set SQL_CONNECT = vtx0_11:scott/tiger
  ```

2. Go to the synergyde\connect\synsqlx directory to compile, link, and run the example programs. There are several example programs (**exam_create_table**, **exam_fetch**, etc.), which are used to test your connection and set-up. Run **exam_create_table** first: it creates a table, which is then used by the other example programs.

   For example:

   ```
   dbl exam_create_table
   dblink exam_create_table
   dbr exam_create_table
   ```

   If this test is unsuccessful, note the error(s) and refer to the "Error Logging and Messages" chapter of the *SQL Connection Reference Manual*. For more information about the example programs, see "Writing an SQL Connection Program" in the "Creating SQL Connection Programs" chapter of the *SQL Connection Reference Manual*.

   Your SQL Connection configuration is now complete. See the *SQL Connection Reference Manual* and the release notes, **REL_CONN.TXT**, for more information about using SQL Connection.

xfODBC: testing the network connection for client access

After configuring SQL OpenNet on the server and installing Connectivity Series or xfODBC Client on the client, you should test the network connection before attempting to access data using xfODBC. To do this, run **vtxping** on the client machine:

```
vtxping [-pport] server_name
```

where *port* is the port number on which the client will connect to SQL OpenNet, and *server_name* is the host name of the SQL OpenNet machine. If no port is specified, the default port specified in the *services* file is used.

If the network connection is working properly, you'll receive a "vtxnetd is alive and kicking" message. If there's a problem, you'll see an error message. See "The vtxping Utility" on page 4-32 for more information.

We recommend that you complete the tutorial to ensure that you can connect to a database and to learn more about using xfODBC. See "Using the Sample Database as a Tutorial" in the *xfODBC User's Guide*.

xfODBC: stand-alone access

After installing Connectivity Series, we recommend that you complete the tutorial to ensure that you can connect to a database and to learn more about using xfODBC. See "Using the Sample Database as a Tutorial" in the *xfODBC User's Guide*.

The sqld program

The **sqld** program provides the Synergy/DE SQL OpenNet service. It reads **opennet.srv** and starts the **vtxnetd/vtxnet2** server program (see “Customizing the opennet.srv file” on page 4-9). **Sqld** also polls **vtxnetd/vtxnet2** periodically to verify that the program is still running (see “Changing the polling interval” on page 4-9). The service name for **sqld** is **SynSQL**; the display name is “Synergy/DE OpenNet Server”.

⚠ Do not attempt to issue **sqld** commands from the command line while the Synergy Configuration Program is running.

Syntax

```
sqld [-h] [-l] [-q|-r|-rs|-x]
```

Arguments

-h

Display a list of **sqld** options.

-l

Log debug information about **SynSQL** startup in the Windows event log. Used with **-r** or **-rs**.

-q

Stop the service **SynSQL**. (See “Stopping and removing SQL OpenNet” on page 4-8 for details on what happens when the service is stopped.)

-r

Register the service **SynSQL**.

-rs

Register the service **SynSQL** (if necessary) and then start it.

-x

Unregister (remove) the service **SynSQL**. If the service is running, **-x** will first stop it.

SQL Connection and xfODBC on UNIX

This section explains how to configure SQL Connection and xfODBC for remote use with SQL OpenNet and how to configure SQL Connection for local (stand-alone) use.

- To run **SQL Connection in a client/server configuration**, see "SQL Connection: using SQL OpenNet" on page 4-14 for server set-up and "SQL Connection: configuring client or stand-alone access" on page 4-17 for client set-up.
- To run **SQL Connection stand-alone**, see "SQL Connection: configuring client or stand-alone access" on page 4-17. (Stand-alone is defined as any connection that doesn't use SQL OpenNet.)
- To run **xfODBC in a client/server configuration**, see "xfODBC: using SQL OpenNet" on page 4-19 for UNIX server set-up and "xfODBC: testing the network connection for client access" on page 4-12 for Windows client set-up.

To use Connectivity Series products, you must source the **setsde** script. This script enables the Synergy/DE environment for the products you have installed. See "Environment setup" on page 1-6 for additional information. The **setsde** script can be used with any UNIX shell *except* **csh**.

Understanding SQL OpenNet on UNIX

SQL OpenNet consists of a daemon process (**vtxnetd**), which listens for requests and starts child processes to service requests from remote clients. The driver name (e.g., **VTX0**) specified in the connect string determines the executable file that the service manager uses to perform the actual servicing of requests. It uses TCP/IP for communication.

See "The vtxnetd and vtxnet2 Programs" on page 4-27 for more information.

SQL Connection: using SQL OpenNet

To use SQL OpenNet for SQL Connection access, you need to

- configure SQL OpenNet by setting the port and building a shared library specific to your database. SQL OpenNet includes two prebuilt shared libraries: **VTX3.so**, which is used for network connections, and **VTX4.so**, which is used for Synergy database connections.
- start SQL OpenNet.

Configuring SQL OpenNet for SQL Connection

You should ensure that the following requirements are met before configuring SQL OpenNet:

- TCP/IP must be installed, configured, and working properly.
- You must have the **ld** C linker (required by **makessqlsrv**).

- On most UNIX platforms, you need a C compiler (required by **makessqlsrv**). However, there are some combinations of UNIX platform and database that do not require a C compiler. The GCC compiler can be used only on Linux.
- (Oracle only) The Oracle Call Interface (OCI) must be installed.
- (Oracle only) You must source the Oracle script **oraenv**.

1. Install Connectivity Series from your distribution and source the **setsde** script (see the note on page 4-14).
2. Set the port number by adding the following line to your system's **/etc/services** file:

   ```
   vtxnet nnnn/tcp #Synergy/DE SQL OpenNet server
   ```

   where *nnnn* is the port number for SQL OpenNet. The default port number is 1958, but you can specify any unused port.

   ⚠ If you use a non-default port, you must also do the following:

   - Specify the port in the **startnet** file. See "Starting SQL OpenNet" on page 4-16.
   - Specify the port on the client side—either in the client's services file or in the connect string. See "Setting the port number (for client access)" on page 4-17.

   You can override the port number in the **services** file by specifying the port on the command line when running **vtxnetd**. See "The vtxnetd and vtxnet2 Programs" on page 4-27 for details on the **vtxnetd** command-line options.
3. Build the server shared library corresponding to the program that your client application will specify in the connect string. For example, for Oracle, you would build the shared library **VTX0.so**, which corresponds to the **VTX0** database driver. The shared library files (other than **VTX3.so** and **VTX4.so**) must be built at install time for the specific version of your database. They should always be built on the target machine; do not build them on one machine and then transfer them to another machine.

   - Move to the SQL Connection directory (below /synergyde/connect). For example:

     ```
     cd synsqlx
     ```

   - Run the **sqlunixbld** program to customize the build process for your database:

     ```
     sqlunixbld
     ```

     The **sqlunixbld** program will prompt you to select the database and version you're using. It uses this information to build the **lib/ssql_libs** file, which is used by **makessqlsrv** to create the shared library.

   - Run **makessqlsrv** to create a new shared library (**VTX*n*.so**) in accordance with the customization you specified when running **sqlunixbld**:

     ```
     makessqlsrv
     ```

4. Move the newly-built shared library to the /synergyde/connect directory.
5. SQL OpenNet is now configured. You must license your database drivers before you can run the example programs. See chapter 2, "Configuring License Manager," for licensing information.

Starting and stopping SQL OpenNet for SQL Connection

Starting SQL OpenNet

Follow these instructions to configure and source the **startnet** script, which kills and restarts SQL OpenNet, and can also be used to set SQL Connection environment variables.

> If you will be connecting to an Oracle database, you must source the Oracle script **oraenv** before starting SQL OpenNet—even if you've already sourced it to configure SQL OpenNet.

1. Move to the connect directory. For example:

```
cd /usr/synergyde/connect
```

2. (optional) Edit the set-up script file, **startnet**, to include SQL Connection environment variables. For instructions, see "Using network initialization files to set network defaults" in the "Welcome to SQL Connection" chapter of the *SQL Connection Reference Manual.*
3. If you are using a port number other than the default (1958), change the port number in all three places it appears in the **startnet** file.
4. Source the set-up script file:

```
. ./startnet
```

   If the server starts successfully, you'll receive a "vtxnetd is alive and kicking" message.

5. To run SQL OpenNet at system start-up, add the **setsde** and **startnet** scripts to your start-up file, for example **/etc/rc**. Make sure these scripts are sourced *after* License Manager is started.

   If you do not want to use **startnet**, you can start SQL OpenNet by typing the **vtxnetd** command on the command line:

   - To start the server and keep it running after you log out, enter

```
nohup vtxnetd &
```

   - To start the server and terminate it when you log out, enter

```
vtxnetd &
```

   See "The vtxnetd and vtxnet2 Programs" on page 4-27 for details on the **vtxnetd** command-line options. After executing the **vtxnetd** command, we recommend that you use **vtxping** to verify that the server is running. See "The vtxping Utility" on page 4-32 for more information.

Stopping SQL OpenNet

To stop SQL OpenNet, use **vtxkill**. Running **vtxkill** kills the daemon process so that no new connections can be made, but it does not terminate existing connections. The syntax is

```
vtxkill [-pport] server_name
```

where *port* is the port number that the server is running on, and *server_name* is the host name of the SQL OpenNet machine. If no port is specified, the default port specified in the **etc/services** file is used.

SQL Connection: configuring client or stand-alone access

To use SQL Connection for client access to a remote database or for stand-alone access to a local database, you need to

- install Connectivity Series and source the **setsde** script (see "Environment setup" on page 1-6).
- set the port number (client only).
- build the shared library for the database (stand-alone only).
- test SQL Connection using the example programs.

Setting the port number (for client access)

Set the port number by either adding a line to the **/etc/services** file or specifying the port in the connect string.

- To set the port number in the **/etc/services** file, add this line:

  ```
  vtxnet nnnn/tcp #Synergy/DE SQL Connection
  ```

  where *nnnn* is the port number. The default port is 1958. If you configured SQL OpenNet to run on a different port, specify that port number here.

- To specify the port number in the connect string see %SSC_CONNECT in the "Database Functions" chapter of the *SQL Connection Reference Manual*. The port in the connect string overrides the port setting in the **services** file.

Building a shared library (for stand-alone access)

For stand-alone access, you'll need to build an SQL Connection shared library. The shared library should be built on the target machine; do not build it on one machine and then transfer it to another machine. Review the following requirements before building a shared library:

- You must have the **ld** C linker (required by **makessqlsrv**).
- On most UNIX platforms, you need a C compiler (required by **makessqlsrv**). However, there are some combinations of UNIX platform and database that do not require a C compiler. The GCC compiler can be used only on Linux.

- (Oracle only) The Oracle Call Interface (OCI) must be installed.
- (Oracle only) You must source the Oracle script **oraenv**.

1. Move to the SQL Connection directory. For example:

   ```
   cd /usr/synergyde/connect/synsqlx
   ```

2. Run the **sqlunixbld** program to customize the build process for your specific database:

   ```
   sqlunixbld
   ```

   The **sqlunixbld** program will prompt you to select the database and version you're using. It uses this information to build the **lib/ssql_libs** file, which is used by **makessqlsrv** to create the shared library.

3. Run **makessqlsrv** to create a new shared library (**VTX*n*.so**) in accordance with the customization you specified when running **sqlunixbld**:

   ```
   makessqlsrv
   ```

4. Move the newly-built **VTX*n*.so** to the /synergyde/connect directory.

5. SQL Connection is now configured. You must license your database drivers before you can run the sample programs. See chapter 2, "Configuring License Manager," for licensing information.

Testing your connection with the sample programs

1. Verify that the target DBMS server is running. (You may want to test this by using a tool from the database vendor.)

2. Set the SQL_CONNECT environment variable to specify a connection string. The connection string varies depending on the set-up. The examples below use the Bourne shell.

   - For a UNIX client to a Windows, UNIX, or OpenVMS server, use a string similar to

     `SQL_CONNECT=net:`*uid/pwd[/db]@[port:]srv_name*`!`*driver_name* `;export SQL_CONNECT`

   - For stand-alone, use a string similar to

     `SQL_CONNECT=`*driver_keyword*`:`*uid/pwd* `;export SQL_CONNECT`

   Where *uid* is an RDBMS log-in user ID, *pwd* is an RDBMS log-in password, *db* is the name of the database (for SQL Server), *port* is the port number on which SQL OpenNet is running (required for non-default port), *srv_name* is your server machine's unique name, *driver_name* is the database driver (e.g., **VTX0**), and *driver_keyword* is the keyword for the database driver (e.g., oracle). For more information, see "Building Connect Strings" in the "Creating SQL Connection Programs" chapter of the *SQL Connection Reference Manual*.

3. Compile and link the example programs. There are several example programs (**exam_create_table**, **exam_fetch**, etc.) located in the synergyde/connect/synsqlx directory. These programs are for use with MySQL, Oracle, SQL Server, and Synergy databases. Run **exam_create_table** first: it creates a table, which is then used by the other example programs.

For example:

```
dbl exam_create_table
dblink exam_create_table
dbr exam_create_table
```

For more information about the example programs, see "Writing an SQL Connection Program" in the "Creating SQL Connection Programs" chapter of the *SQL Connection Reference Manual*.

4. Run the example programs to test your connection and set-up. If this test is unsuccessful, note the error(s) and refer to the "Error Logging and Messages" chapter of the *SQL Connection Reference Manual*.

   Your Synergy/DE SQL Connection configuration is now complete. See the *SQL Connection Reference Manual* and the release notes, **REL_CONN.TXT**, for more information about using SQL Connection.

xfODBC: using SQL OpenNet

To use SQL OpenNet for xfODBC access, you need to

- configure SQL OpenNet for xfODBC.
- start SQL OpenNet.

Configuring SQL OpenNet for xfODBC

Before configuring SQL OpenNet, you must have TCP/IP installed, configured, and working properly.

1. Install Connectivity Series from your distribution and source the **setsde** script (see "Environment setup" on page 1-6).

2. Add the SQL OpenNet port number to the TCP/IP services file. Open your system's **/etc/services** file and add this line:

   ```
   vtxnet nnnn/tcp #Synergy/DE SQL OpenNet server
   ```

   where *nnnn* is the port number for SQL OpenNet. The default port number is 1958, but you can specify any unused port.

   > If you use a non-default port, you must also specify it on the client side in the ODBC Data Source Administrator dialog box. See "Setting Up Access with DSNs" in the "Configuring Data Access" chapter of the *xfODBC User's Guide*.

   You can override the port number in the **services** file by specifying the port on the command line when running **vtxnetd**. See "The vtxnetd and vtxnet2 Programs" on page 4-27 for details on the **vtxnetd** command-line options.

Starting and stopping SQL OpenNet for xfODBC

Follow these instructions to configure and source the **setodbc** script, which kills and restarts SQL OpenNet, and can also be used to set xfODBC environment variables.

1. Move to the synodbc directory. For example:

   ```
   cd /usr/synergyde/connect/synodbc
   ```

2. (optional) Edit the set-up script file, **setodbc**, to include xfODBC environment variables. See "Setting environment variables for xfODBC" in the "Preliminary Steps" chapter of the *xfODBC User's Guide* for more information.

3. If you are using a port number other than the default (1958), change the port number in all three places it appears in the **startnet** script file. (This file is located in the synergyde/connect directory; it is executed by **setodbc**.)

4. Source the set-up script file:

   ```
   . ./setodbc
   ```

   If the server starts successfully, you'll receive a "vtxnetd is alive and kicking" message.

   We do not recommend using the DBLCASE environment variable with xfODBC. However, if you do use it, and you see this error when the **setodbc** script runs

   ```
   Cannot open /usr/synergyde/connect/synodbc/GENESIS.ISM
   ```

   it is likely that DBLCASE was not set when **setodbc** and **dbcreate** were initially run. On this initial run, there was no problem. However, when the system was rebooted, it is likely that DBLCASE was set to **u:l**, and then when **setodbc** subsequently ran, there was a case mismatch, resulting in an error.

   To correct this situation, set DBLCASE to blank and source **setodbc**. To avoid seeing this error in the future, DBLCASE must be set to blank whenever **dbcreate** is run and SQL OpenNet or a local ODBC application is started.

5. To run SQL OpenNet at system start-up, add the **setsde** and **setodbc** scripts to your start-up file, for example **/etc/rc**. Make sure the **setsde** and **setodbc** scripts are sourced *after* License Manager is started.

   Your xfODBC server component is now ready to use. For information about using xfODBC, see the *xfODBC User's Guide* and the online release notes file, **REL_CONN.TXT**.

   If you do not want to use **setodbc**, you can start SQL OpenNet by typing the **vtxnetd** command on the command line:

   - To start the server and keep it running after you log out, enter

     ```
     nohup vtxnetd &
     ```

   - To start the server and terminate it when you log out, enter

     ```
     vtxnetd &
     ```

See “The vtxnetd and vtxnet2 Programs” on page 4-27 for details on the **vtxnetd** command-line options. After executing the **vtxnetd** command, we recommend that you use **vtxping** to verify that the server is running. See “The vtxping Utility” on page 4-32 for more information.

Stopping SQL OpenNet

To stop SQL OpenNet, use **vtxkill**. Running **vtxkill** kills the daemon process so that no new connections can be made, but it does not terminate existing connections. The syntax is

```
vtxkill [-pport] server_name
```

where *port* is the port number that the server is running on, and *server_name* is the host name of the SQL OpenNet machine. If no port is specified, the default port specified in the **etc/services** file is used.

SQL Connection and xfODBC on OpenVMS

This section explains how to configure SQL Connection and xfODBC for remote use with SQL OpenNet and how to configure SQL Connection for local (stand-alone) use.

- To run **SQL Connection in a client/server configuration**, see “SQL Connection: using SQL OpenNet” on page 4-22 for server set-up and “SQL Connection: configuring and testing client or stand-alone access” on page 4-24 for client set-up.
- To run **SQL Connection stand-alone**, see “SQL Connection: configuring and testing client or stand-alone access” on page 4-24. (Stand-alone is defined as any connection that doesn’t use SQL OpenNet.)
- To run **xfODBC in a client/server configuration**, see “xfODBC: using SQL OpenNet” on page 4-25 for OpenVMS server set-up and “xfODBC: testing the network connection for client access” on page 4-12 for Windows client set-up.

To use Connectivity Series products, you must execute the command file **SYS$MANAGER:SYNERGY_STARTUP.COM**, which executes **SYS$MANAGER:CONNECT_STARTUP.COM** to set up the environment for Connectivity Series. **SYS$MANAGER:SYNERGY_STARTUP.COM** should be added to your OpenVMS system start-up procedure; refer to the installation instructions that came with your distribution (they are also available on the Synergex website) for more information.

Understanding SQL OpenNet on OpenVMS

SQL OpenNet runs as the **vtxnetd** detached process, which listens for requests and starts child processes to service requests from remote clients. Each database driver consists of **VTX*n*.EXE** and an associated shared image, **VTX*n*_SO.EXE**. The driver name (e.g., **VTX0**) in the connect string determines the executable file and its associated shared image that the service manager uses to perform the actual servicing of requests. It uses TCP/IP for communication.

See “The vtxnetd and vtxnet2 Programs” on page 4-27 for more information.

SQL Connection: using SQL OpenNet

To use SQL OpenNet for SQL Connection access, you need to

- configure and build SQL OpenNet for SQL Connection. SQL OpenNet includes two prebuilt database drivers: **VTX3_SO.EXE**, which is used for network connections, and **VTX4_SO.EXE**, which is used for Synergy database (RMS) connections.
- start SQL OpenNet.

Configuring and building SQL OpenNet for SQL Connection

1. Install Connectivity Series from your distribution and run the command procedure **SYS$MANAGER:SYNERGY_STARTUP.COM** (see the note on page 4-22).

2. Use SET DEF to move to the location of the SQL OpenNet directory. For example:

```
$ SET DEF CONNECTDIR:
```

3. Build the SQL Connection database driver corresponding to the program that your application will specify in the connect string. For example, for Oracle you would build the driver **VTX0_SO.EXE**. If your client application will be connecting to a Synergy database, you do not need to build a driver. Instead, use the prebuilt database driver **VTX4_SO.EXE**.

```
$ @BUILD_SSQL_DB rdbms
```

where *rdbms* is one of the following: ORACLE10, SYBASE-DB, RDB7.

If you are building an SQL Connection database driver for an Oracle driver, the Oracle Call Interface (OCI) must be installed.

4. (optional) To change the default port number (1958), modify the **vtxnetd** line in **NET.COM**. For example, to use port 1960 you'd enter

```
$ VTXNETD -p1960
```

When a client connects to SQL OpenNet, the client application specifies the port number it will use. If you change the default port in **NET.COM**, you must also change the port settings in the connect strings on all clients (see "SQL Connection: configuring and testing client or stand-alone access" on page 4-24). For details on the **vtxnetd** command-line options, see "The vtxnetd and vtxnet2 Programs" on page 4-27.

Starting and stopping SQL OpenNet for SQL Connection

Starting SQL OpenNet

If you use database logicals, be sure to define them in your system start-up command procedure before starting SQL OpenNet.

1. Use SET DEF to move to the location of the SQL OpenNet directory, and then execute the SQL OpenNet start-up command file. For example:

```
$ SET DEF CONNECTDIR:
$ @STARTNET
```

We recommend starting SQL OpenNet from your OpenVMS system start-up file.

2. Verify that the server is running:

   ```
   $ VTXPING [-pport] server_name
   ```

   where *port* is the port number that the server is running on and *server_name* is the host name of the SQL OpenNet machine. If no port is specified, 1958 is used.

   If the server started successfully, you'll receive a "vtxnetd is alive and kicking" message. If there's a problem, you'll see an error message. See "The vtxping Utility" on page 4-32 for more information. For SQL Connection, you must license your database drivers before you can run the example programs. See chapter 2, "Configuring License Manager," for licensing information.

Stopping SQL OpenNet

To stop SQL OpenNet, use **vtxkill**. Running **vtxkill** disables new connections, but does not terminate existing processes. **Vtxnetd** stops only when all children have terminated. The syntax is

```
$ VTXKILL [-pport] server_name
```

where *port* is the port number that the server is running on, and *server_name* is the host name of the SQL OpenNet machine. If no port is specified, the default port specified in the **NET.COM** file is used.

SQL Connection: configuring and testing client or stand-alone access

After you install Connectivity Series and run **SYS$MANAGER:SYNERGY_STARTUP.COM**, no additional configuration is necessary for client access to a remote database; start with step 2 below to test your installation. For stand-alone access, you will need to first build the driver before testing the installation.

1. (Stand-alone access only) Build the SQL Connection database driver that your application will use following the instructions in step 3 on page 4-23.
2. Set the SQL_CONNECT environment variable to specify a connection string. The connection string varies depending on the set-up.

   For example:

   - For an OpenVMS client to a Windows, UNIX, or OpenVMS server, use a string similar to

     ```
     $ DEFINE SQL_CONNECT net:uid/pwd[/db]@[port:]srv_name!driver_name
     ```

   - For stand-alone, use a string similar to

     ```
     $ DEFINE SQL_CONNECT driver_keyword:uid/pwd
     ```

   Where *uid* is an RDBMS log-in user ID, *pwd* is an RDBMS log-in password, *db* is the name of the database (for SQL Server), *port* is the port number on which SQL OpenNet is running (required for non-default port), *srv_name* is your server machine's unique name, *driver_name* is the driver name (e.g., **VTX0**), and *driver_keyword* is the keyword for the driver (e.g., oracle). For more information, see "Building Connect Strings" in the "Creating SQL Connection Programs" chapter of the *SQL Connection Reference Manual*.

3. Compile, link, and run the example programs. There are several example programs (EXAM_CREATE_TABLE, EXAM_FETCH, etc.) located in the directory DEVICE:[SYNERGYDE.CONNECT.SYNSQLX]. These programs are for use with MySQL, Oracle, SQL Server, and Synergy databases. Run EXAM_CREATE_TABLE first: it creates a table, which is then used by the other example programs.

   For example:

   ```
   $ DBL EXAM_CREATE_TABLE
   $ LINK EXAM_CREATE_TABLE, sys$share:ssqlrtl/opt
   $ RUN EXAM_CREATE_TABLE
   ```

   For more information about the example programs, see "Writing an SQL Connection Program" in the "Creating SQL Connection Programs" chapter of the *SQL Connection Reference Manual*.

   > Use the link options file **ssqlrtl.opt** (instead of **synrtl.opt**) to link an SQL Connection application.

4. Test the SQL Connection runtime image by running your SQL Connection application and trying to access data from the target database. If this test is unsuccessful, note the error(s) returned and refer to the "Error Logging and Messages" chapter of the *SQL Connection Reference Manual*.

   See the *SQL Connection Reference Manual* and the release notes, **REL_CONN.TXT**, for more information about using SQL Connection.

xfODBC: using SQL OpenNet

To use SQL OpenNet for xfODBC access, you need to

- configure SQL OpenNet for xfODBC (optional).
- start SQL OpenNet.

Configuring SQL OpenNet for xfODBC

Follow these instructions to *change the default port number* that SQL OpenNet uses for RMS file access. If you want to use the default port (1958), *you do not need to read this section*; go directly to "Starting and stopping SQL OpenNet for xfODBC" on page 4-26.

> If you have not logged out between installing Connectivity Series and configuring SQL OpenNet, the SQL OpenNet start-up command procedure is already running, and you can skip step 1.

1. Run the SQL OpenNet start-up command procedure:

   ```
   $ @SYS$MANAGER:CONNECT_STARTUP
   ```

2. Use SET DEF to move to the location of the SQL OpenNet directory. For example:

   ```
   $ SET DEF CONNECTDIR:
   ```

3. To change the default port (1958), modify the **vtxnetd** line in the input file **NET.COM**. For example, to use port 1960 you'd enter

   ```
   $ VTXNETD -p1960
   ```

When a client connects to SQL OpenNet, the client application specifies the port number it will use. If you change the default port in the input file, you must also change the port number settings in the connect strings on all clients to match; see "Setting Up Access with DSNs" in the "Configuring Data Access" chapter of the *xfODBC User's Guide*. For details on the **vtxnetd** command-line options, see "The vtxnetd and vtxnet2 Programs" on page 4-27.

Starting and stopping SQL OpenNet for xfODBC

Starting SQL OpenNet

1. Use SET DEF to move to the location of the SQL OpenNet directory, and then execute the SQL OpenNet start-up command file. For example:

   ```
   $ SET DEF CONNECTDIR:
   $ @STARTNET
   ```

2. Verify that the server is running:

   ```
   $ VTXPING [-pport] server_name
   ```

   where *port* is the port number that the server is running on, and *server_name* is the host name of the SQL OpenNet machine. If no port is specified, 1958 is used.

   If the server started successfully, you'll receive a "vtxnetd is alive and kicking" message. If there's a problem, you'll see an error message. See "The vtxping Utility" on page 4-32 for more information.

Stopping SQL OpenNet

To stop SQL OpenNet, use **vtxkill**. Running **vtxkill** disables new connections, but does not terminate existing processes. **Vtxnetd** stops only when all children have terminated. The syntax is

```
$ vtxkill [-pport] server_name
```

where *port* is the port number that the server is running on, and *server_name* is the host name of the SQL OpenNet machine. If no port is specified, 1958 is used.

The vtxnetd and vtxnet2 Programs

Vtxnetd and **vtxnet2** are service programs that implement SQL OpenNet. **Vtxnet2** is available only on Windows. Note the following operating system–specific information:

- On Windows, the command line that starts **vtxnetd** or **vtxnet2** is in the **opennet.srv** file. **Vtxnetd** is the default. See "Customizing the opennet.srv file" on page 4-9.
- On UNIX, the command line to start **vtxnetd** is located in the **startnet** script. **Vtxnetd** can also be run from the command line. We don't recommend running **vtxnetd** with root privileges or uid=0 (unless you are using the **-a** option). See "Starting and stopping SQL OpenNet for xfODBC" on page 4-20.
- On OpenVMS, the line for starting the database driver is located in the input file **NET.COM** (see page 4-26).

If the SQL OpenNet service fails to start, an error will be written to the **tcm_*pid*.log** file if you are using **vtxnetd/vtxnet2** logging. There will be no other indication that the service failed to start, except that SQL OpenNet connections will fail.

Syntax

`vtxnetd` *[-option] [...]*

or

`vtxnet2` *[-option] [...]*

Arguments

option

One or more of the following:

-a[L] Use the operating system to validate the user name and password in connect strings. On Windows, you can use **-a** with the **L** option (that is, **-aL**; **L** must be capitalized) to log authorization errors to the Windows Event Viewer.

-e*crt key [p]* Use SSL data packet encryption. *Crt* is the name and location of the SSL certificate file, *key* is the name and location of the private key file, and *p* (optional) specifies supported protocols (TLS levels).

-f*filename* Use the specified file to check inbound service requests. (Windows, UNIX only)

-h Display a list of options.

-k*n* Encrypt user names and passwords in connect strings; *n* specifies the key for the encryption algorithm.

log Output a connection log.

log2 Output a connection log that does not include **vtxping** events.

-p*nnnn* Listen for requests on port *nnnn*. (Windows, UNIX only)

-s*n* Specify thread stack size for **vtxnetd** in kilobytes. (Windows only)

-v6 Specify that the port allows IPv6 connections. Default is IPv4. If the OS supports IPv4 and IPv6 connections on the same port, specifying **-v6** means both IPv4 and IPv6 are supported. If the OS *does not* support IPv4 and IPv6 on the same port, specifying **-v6** results in *only* IPv6 support. (Currently, only Solaris and HP-UX do not offer this support.) (Windows, UNIX only)

-w*n* Terminate existing connections when **vtxnetd** is shut down. There is a delay of up to *n* seconds after the shut down is requested for underlying operations to complete. The default is 10 seconds. (Windows only)

Discussion

Starting the service manager with a password (-a[L])

By default, no system-level remote user authentication occurs. This behavior can be overridden with the **-a** option. When **-a** is used, the user name and password for the host must be passed in the connect string from the client machine (in addition to any database user name and password). The host user name and password could be an account on the server machine or (on Windows) on a domain controller.

The service manager validates the user name and password using operating system security validation. Assuming the user name and password are valid, the service manager creates a new process or thread for the connection using the persona of the user name that was passed to it. Note the following operating system restrictions for using the **-a** option:

- On Windows, if you are using **vtxnet2**, the account for this user must have the “log on as a batch job” user privilege.
- On UNIX, **vtxnetd** must be started by a user with root privileges or uid=0. Otherwise, we do not recommend running **vtxnetd** as root or uid=0.
- On OpenVMS, the requested driver must be started with the system user ID [1,4].

On Windows, by including “**L**” with the **-a** option (that is, **-aL**), you can log authorization errors in the Windows application event log, which can be viewed with the Windows Event Viewer. If the authentication fails, this log will contain information that may be helpful in troubleshooting.

See “Building Connect Strings” in the “Creating SQL Connection Programs” chapter of the *SQL Connection Reference Manual* and “Setting Up Access with DSNs” in the “Configuring Data Access” chapter of the *xfODBC User’s Guide* for more details on connection strings and specifying the optional user name and password.

Encrypting data packets (-e)

By default, no data packet encryption occurs for SQL OpenNet. To enable data packet encryption, use the **-e** option:

`-e` *crt key [p]*

This option invokes SSL encryption functionality supplied by a third-party library, OpenSSL. *Crt* specifies the name and location of the SSL certificate file, and *key* specifies the name and location of the private key file. These can include a full path or logical. The optional *p* argument enables you to specify the supported TLS levels (TLS 1.0, 1.1, 1.2). By default, all three are supported. To specify more than one TLS level, separate values with commas. For example,

```
vtxnetd -e c:\ssl\crt.txt c:\ssl\key.txt 1.1,1.2
```

See “Using Data Packet Encryption for SQL OpenNet” on page 4-3 for more information.

Rejecting invalid service requests (-f)

The **-f** option is supported on Windows and UNIX only.

Using the **-f** option tells the service manager to check inbound service requests and reject invalid requests. The **-f** option specifies a text file, in which you list the connect strings that this instance of the service manager can connect to. For example, if the file of valid connection strings contains only this:

```
/usr2/synergyde/connect/vtx0
```

then only a connection request from a client specifying exactly

```
!/usr2/synergyde/connect/vtx0
```

will be allowed.

Encrypting user names and passwords (-k)

By default, the **-k** option is included on the command line in the start-up file. This option encrypts both the database user name and password and the host user name and password being sent across the wire.

The key *n* can be any number between 1 and 2,147,483,647. It must be set to the same value on both the server and the client. On the client, set it in the **net.ini** file in the connect\synodbc\lib directory. See “Setting connect string defaults and encryption in net.ini” in the “Welcome to SQL Connection” chapter of the *SQL Connection Reference Manual*. If these numbers do not match, you’ll get the error “Invalid connect syntax (uid/pwd/datasource).”

Using a log file (log, log2)

Use the **log** option to produce a log file that contains error information and connection requests, including **vtxping** events, such as occur when **sqld** polls **vtxnetd/vtxnet2**. Use the **log2** option to produce a log file that contains error information and connection requests, without the **vtxping** events. Using **log2** produces a smaller logfile. By default, the log file is located in the directory that the service manager was started in and is named **tcm_*pid*.log**, where *pid* is the current process ID.

Other types of logging are available; see "SQC Connection Troubleshooting and Error Logging" in the "Error Logging and Messages" chapter in the *SQL Connection Reference Manual*. For information on xfODBC logging, see "Error Logging for xfODBC" in the "Data Access Errors and Error Logging" chapter of the *xfODBC User's Guide*.

Overriding the default port number (-p)

The default port (1958) used by the service manager is defined in the **services** file (on Windows and UNIX) and in the **NET.COM** file on OpenVMS. (For details on setting the port number in **services** or **NET.COM**, refer to the operating system–specific sections of this chapter.) On Windows and UNIX, the default port can be overridden with the **-p** option; this option is not supported on OpenVMS.

If you use a non-default port for **vtxnetd/vtxnet2**, you must also change the port number on the client. For SQL Connection, For SQL Connection, see page 4-10 (Windows), page 4-17 (UNIX), or page 4-24 (OpenVMS). For xfODBC, see "Setting Up Access with DSNs" in the "Configuring Data Access" chapter of the *xfODBC User's Guide*.

Specifying the amount of memory used by vtxnetd on Windows (-s)

By default, **vtxnetd** uses 512K of memory on Windows. You can change this with the **-s** option, which determines how much memory **vtxnetd** will use by limiting the thread stack size allocated to it. Reducing the amount of memory used may enable **vtxnetd** to support more concurrent users.

For example, to reduce the amount of memory used by **vtxnetd** to 256K, you would use the following:

```
vtxnetd -s256
```

Database drivers have minimum requirements for thread stack size. If you set the thread stack size to a value that is less than the minimum required by your database driver, **vtxnetd** will likely fail or generate random errors.

Shutting down vtxnetd on Windows (-w)

By default, the **-w** option is included on the command line in the **opennet.srv** file. When **-w** is specified, shutting down SQL OpenNet will disable new connections and terminate existing ones, resulting in a "Network connection lost" (10054) error on the client. There is a delay of up to *n* seconds before existing connections are terminated, allowing underlying operations to complete.

The default is 10 seconds, but you can change this value by editing the **-w** option on the **vtxnetd** command line in **opennet.srv**. If you set it to 0 (zero), all connections will be terminated immediately.

If **-w** is not specified, new connections are disabled, but existing connections are not terminated, and **vtxnetd** does not completely shut down until all child processes have terminated. Only the third-party applications that are connected to the Synergy data can stop existing processes in this case.

See "Stopping and removing SQL OpenNet" on page 4-8 for details on the various ways to stop the service.

The **-w** option was introduced in version 10.3 and is included on the **vtxnetd** command line in the **opennet.srv** and **opennet_base.srv** files for 10.3 and higher. The **opennet.srv** file is not overwritten on an upgrade, however, so if you are upgrading from a previous version and want client connections to be terminated when **vtxnetd** is shut down, you will need to edit **opennet.srv** to add **-w***n* to the command line. For example,

```
vtxnetd.exe -k67834 -p1958 -s512 -w10 log2
```

Running multiple SQL OpenNet servers

You can run multiple SQL OpenNet servers by specifying multiple start-up lines, with a different port number for each server. For example:

- On Windows, add the additional start-up lines to the **opennet.srv** file:

```
vtxnetd.exe -p1960
```

- On UNIX, add the additional start-up lines to the **startnet** script file:

```
nohup vtxnetd -p1960 &
```

- On OpenVMS, add the additional start-up lines to the **NET.COM** file:

```
$ VTXNETD -p1960
```

The vtxping Utility

This program is used to test a connection to SQL OpenNet. **Vtxping** uses *only* TCP/IP API calls and tests *only* whether the server has been started successfully. **Vtxping** prints a report to the screen. If there are errors, the report includes the TCP/IP error number. This information can be used by your network administrator to resolve TCP/IP network socket communication problems.

For more verbose error reporting, run **synxfpng** with the **-x** and **-v** options. See "The synxfpng Utility" on page 3-58.

Syntax

vtxping *[-m*num*] [-p*port*] server_name [server_name…]*

Arguments

-m*num*

(optional) Retry a maximum of *num* times. The default is 3.

-p*port*

(optional) Ping the specified port (the port number on which SQL OpenNet is running). Required if you are using a non-default port.

server_name […]

The host name of the machine that SQL OpenNet is running on. You can put multiple server names on the command line as long as they all use the same port number. **Vtxping** will ping each one in turn.

Discussion

If **-p** is not specified, **vtxping** uses the port specified in the **services** file on Windows and UNIX and port 1958 on OpenVMS.

When you run **vtxping** without any options, it displays usage information.

Usage

- Ping the SQL OpenNet server named tiger on the default port.

  ```
  vtxping tiger
  ```

- Ping the servers tiger and lion, both of which are running SQL OpenNet on port 4444. If **vtxping** cannot connect on the first try, it will attempt twice more.

  ```
  vtxping -m2 -p4444 tiger lion
  ```

If the server was started successfully, you'll receive a "vtxnetd is alive and kicking" message. If there's a problem, you'll see an error message.

5

Redistributing Synergy/DE on Windows

Redistribution Requirements

When you are ready to distribute your Synergy application, you'll also need to distribute parts of Synergy/DE with it. At a minimum, you need to distribute core components, which include License Manager and the Synergy Runtime; your application may require that you distribute other components or products as well. You can, of course, simply install the Synergy components that your application requires from the web download. But you also have the option of installing Synergy products from the command line. You can install the following products in this way:

- Synergy/DE
- xfNetLink Java and xfNetLink .NET
- xfODBC Client
- Synergy/DE Client
- Synergy DBL Integration for Visual Studio
- Online Manuals

Microsoft .NET Framework must be installed *before* you can install Synergy/DE. For detailed .NET requirements, see "Microsoft .NET Framework requirements" on page 1-2.

For information on installing Synergy/DE Client, refer to the installation instructions that accompanied your distribution.

For detailed information on deploying xfNetLink applications, refer to the section of the *xfNetLink & xfServerPlus User's Guide* that applies to your client.

For detailed information on deploying xfODBC applications, see chapter 4, "Configuring Connectivity Series," in this manual and "xfODBC requirements and installation" in the "Welcome to xfODBC" chapter of the *xfODBC User's Guide*.

The Synergy installations support the Windows Installer command line switch **/q** and its variants, which enable you to limit the number of dialog boxes that the user sees and interacts with during the installation. Using **/q**, you can take advantage of the following user interface levels:

- **Full user interface.** Users see all the available dialog boxes. This is the default.
- **Basic user interface.** Users see a small progress dialog box. You can optionally display a modal status dialog box at the end of the installation. By default, a Cancel button displays on the progress dialog box, but you can choose to hide it.
- **Reduced user interface.** Users see a full-sized progress dialog box with a Cancel button. There is no status dialog box displayed at the end of the installation.
- **No user interface.** Users see no dialog boxes at all; also referred to as a silent installation. You can optionally display a modal status dialog box at the end of the installation.

When performing a basic, reduced, or no user interface (UI) installation, the dialog boxes for license configuration and component selection do not display.

- For license configuration, the existing information is used on an upgrade and on any system on which Synergy/DE was previously installed. On a new installation, you use properties to specify the information. See "Properties" on page 5-6.
- For component selection, the current values are used on an upgrade. On a new installations, default values are used. See "Default components" on page 5-4.

Regardless of the type of installation, the installed Synergy product is included in the Programs and Features list and can be uninstalled from there or from the command line.

Licensing requirements

A full user interface installation displays the Synergex Synergy/DE Product License Agreement Terms and Conditions (PLA) and requires that users accept the terms and conditions specified therein. However, during a basic, reduced, or no UI installation, the PLA is not displayed, and so you must do one of the following:

- Display the PLA as part of your product installation and require users to accept it. A copy of this document (**license.rtf**) can be found in any Synergy product installation. For example, in a Synergy/DE installation, it is located in Synergex\SynergyDE\dbl.
- Require users to sign a paper copy of the PLA.

Installation location

The table below shows the installation locations for the various Synergy products.

Product	Installation directory
Synergy/DE	<Program Files>\Synergex\SynergyDE
xfNetLink Java	<Program Files>\Synergex\xfNLJava
xfNetLink .NET	<Program Files>\Synergex\xfNLNET
xfODBC Client	<Program Files>\Synergex\SynergyDE\Connect
Synergy/DE Client	N/A
Synergy DBL Integration for Visual Studio	N/A
Online Manuals	<Program Files>\Synergex\SynergyDE\Manuals

Default components

The following components are installed by default in a new Synergy/DE installation. If you do not wish to install the default components, you must do a full UI installation and select the components you want to install.

- Core Components
- Professional Series Workbench (32-bit only)
- Professional Series Development Environment (64-bit only)
- Connectivity Series
- xfSeries
- Files for shared installation (Windows Server 2008, 2008 R2, 2012, and 2012 R2 only.)

TIP If you do not require some of the default components, but don’t want to do a full UI installation, you can install the default components and then license only the ones you need.

Installation Options

You can install Synergy products by running **setup.exe** or by running Windows Installer (**msiexec.exe**) directly.

Synergy products must be installed from an elevated command prompt. This is done automatically by **setup**, but not by **msiexec**. Using the latter, if you perform a full or no UI installation from a non-elevated command prompt, an error displays and the installation aborts.

All operating system requirements (such as service packs and minimum versions) apply and can halt an installation if they are not met.

Using setup

To install using **setup**, run the **setup.exe** file specific to the product you are installing. **Setup** calls the **.msi** file (e.g., **101sde1031.msi**) to install the product. You can specify the licensing properties and **/q** options on the command line or with the CmdLine entry in the [Startup] section of the **setup.ini** file.

You need the following files, which can be obtained by unpacking the web download:

setup.exe
setup.ini
filename.msi (e.g., **101sde1031.msi**)
data1.cab
splashver.bmp
0x0409.ini

The splash screen displays when using **setup**, regardless of the user interface level specified, but you can turn it off with the **/s** option. (Put **/s** before **/v**.)

Using the Windows Installer directly

To install using the Windows Installer directly, run **msiexec** and specify the **.msi** filename (e.g., **101sde1031.msi**) on the command line, along with the desired **/q** options and licensing properties (if required).

You need the **.msi** file and the **data1.cab** file. The splash screen does not display.

Supported /q options

Full user interface:

/qf Display all dialog boxes. (default for installing)

Basic user interface:

/qb Display a small progress dialog box. (default for uninstalling)

/qb+ Display a small progress dialog box and a status dialog box.

/qb! Display a small progress dialog box and hide the Cancel button. You can also use **/passive**.

Note that you can use the **+** and **!** options together, **/qb+!**.

Reduced user interface:

/qr Display a full-sized progress dialog box.

No user interface (silent):

/qn Display no dialog boxes. You can also use **/quiet** or just **/q**.

/qn+ Display a status dialog box only.

Properties

Properties are used to specify license information for new installations. For a basic, reduced, or no UI installation, properties enable you to pass the license information during installation. For a full UI installation, properties enable you to pass information that you want to display by default in the License Configuration dialog box (but users can override the default values). License properties apply only to Synergy/DE installations.

Properties are case sensitive.

LICENSETYPE. Specify either Client or NonClient (case sensitive).

SERVERNAME. The name of the license server machine if LICENSETYPE is Client. If the license server is running on a non-default port, append a colon followed by the port number to the end of the server name. See Examples below.

LICENSEENAME. The licensee name if LICENSETYPE is NonClient. If the name has spaces in it, put it in double quotes when running **msiexec** directly or running **setup** and using the **setup.ini** file. When running **setup** and specifying options on the command line, use the escape character “\” before the quotes that enclose the licensee name. See Examples below.

If you do not pass license properties (or if you specify incomplete information) to a basic, reduced, or no UI installation that does not already have licensing, Synergy/DE will install, but licensing will be in an uninitialized state. To initialize License Manager, run the Synergy Configuration Program and follow the prompts, or run **lmu -d** from the command line.

However, if you are installing a 32-bit license client on a 64-bit machine that already has 64-bit Synergy/DE installed *and* is designated as a license server, you do not need to pass LICENSETYPE or SERVERNAME. The former will default to Client and the latter to the name of the 64-bit machine.

If you pass license properties when the target system already has license information from a previous installation of Synergy, the passed properties are ignored.

Examples

TIP

When doing a silent install or uninstall, if an error is encountered, the installation may abort without displaying the error to the screen. If this happens, it may be helpful to look at the log file, which is created automatically by Windows Installer Service and placed in the user's %TEMP% directory.

A. This example shows a silent installation of Synergy/DE using **msiexec** with client licensing.

```
msiexec /i 101sde1031.msi /qn LICENSETYPE=Client SERVERNAME=Tiger
```

B. This example shows a silent installation of xfNetLink .NET using **msiexec**. (License properties are not required for an xfNetLink installation.)

```
msiexec /i 101nn1031.msi /qn
```

C. This example shows a basic UI installation of Synergy/DE using **setup**, which hides the Cancel button and displays a status dialog box when the installation is complete. Note that there is no space after the **/v** option and all the options following **/v** are enclosed in double quotes. The server Tiger is on a non-default port.

```
setup /v"/qb+! LICENSETYPE=Client SERVERNAME=Tiger:5555"
```

D. This example shows a reduced UI installation of Synergy/DE using the **setup** file with stand-alone licensing. To pass a licensee name that contains spaces, use the escape character (\) before the double quotes that enclose the name.

```
setup /v"/qr LICENSETYPE=NonClient LICENSEENAME=\"Acme Computers\""
```

E. This example shows the same installation as example D except that we put the command line options in the **setup.ini** file. If the licensee name contains spaces, enclose it in double quotes (no escape character is required in this case). Be sure to use the **setup.ini** file that is included with the product and version you are installing.

In the **setup.ini** file, do this:

```
CmdLine=/qr LICENSETYPE=NonClient LICENSEENAME="Acme Computers"
```

Then, on the command line, just run **setup** without any additional options:

```
setup
```

F. This example shows a silent uninstallation using **msiexec**. This command must be run from the directory where the **.msi** file is located. If desired, you can use the product code instead of the **.msi** filename. (Use the Orca utility to obtain the ProductCode property from the **.msi** file.)

```
msiexec /x 101sde1031.msi /qn %temp%\uninstall.log
```

G. This example shows a silent uninstallation using **setup**. This command must be run from the directory where the **.msi** file is located. You cannot use **setup.ini** when uninstalling.

```
setup /v"/x 101sde1031.msi /qn %temp%\uninstall.log"
```

Index