Installation Configuration Guide

Version 9.3

SYNERGY DE™

Printed: December 2009

DCN IC-01-9301

Synergex
2330 Gold Meadow Way
Gold River, CA 95670 USA

http://www.synergex.com
phone 916.635.7300
fax 916.635.6549

Contents

Preface

1 Requirements and Considerations

2 Configuring License Manager

3 Configuring *xf*Server

5 Redistributing Synergy/DE on Windows

Index

Preface

About this manual

The *Installation Configuration Guide* focuses on the installation and set-up of the Synergy/DE™ License Manager, *xf*Server, 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 the above products. How-to information for implementing your Synergy™ applications is discussed in *Getting Started with Synergy/DE*.

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 useful publications

- *Getting Started with Synergy/DE*
- *Synergy Language Reference Manual*
- *Environment Variables and System Options*
- *Synergy Language Tools*
- *SQL Connection Reference Manual*
- *xfODBC User's Guide*
- *Developing Distributed Synergy Applications: Using xfNetLink and xfServerPlus*

Product support information

If you cannot find the information you need in this manual or in the publications listed above, you can call Synergy/DE Developer Support at (800) 366-3472 (in North America) or (916) 635-7300. To purchase Synergy/DE Developer Support services, contact your Synergy/DE account manager at the above phone 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 North America) 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 9.3 is supported on the following Windows platforms. For all platforms, both 32-bit and 64-bit are supported, with the exception of Windows XP, which is 32-bit only.

- Windows 7
- Server 2008 R2
- Server 2008 SP2
- Vista SP2
- Server 2003 SP1
- XP SP3

Patches

Regardless of Windows platform, you should always load all Windows critical updates. You can run the **syncksys** program in the synergyde\dbl\bin directory to check for required patches and settings. Detailed information about the messages returned by **syncksys** is available on the **Windows Diagnostic page** in the Resource Center of the Synergex web site; you can also download the latest version of **syncksys** from the Resource Center.

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 and System Options*. For information on setting SFWINIPATH, see SFWINIPATH in that same chapter.

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.

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 on page 1-2.

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

- XP
- Vista
- Windows 7

Note that the client cannot be a Terminal Services machine, which includes Windows Server 2003, 2008, and 2008 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 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 workstation. License configuration keys must be installed on each workstation.

Network licensing uses License Manager server to manage licenses for multiple workstations (license clients). With network licensing, you are required to install configuration keys only on the license server machine. The individual workstations 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 9.3 is supported on the following UNIX platforms.

Platform	System code	Version
HP Tru64	021	5.1b or higher
HP-UX 32-bit (PA-RISC)	009	11i v2 (B.11.23) or higher
HP-UX 64-bit (PA-RISC)	309	11i v2 (B.11.23) or higher
HP-UX 64-bit (Itanium)	509	B.11.23 or higher
IBM AIX 5L 32-bit	004	5.2 or higher
IBM AIX 5L 64-bit	304	5.2 or higher
Linux (x86)	028	RHEL 4 or higher SUSE 9.0 or higher
Linux (x64)	428	RHEL 4 or higher SUSE 9.0 or higher
SCO OpenServer	003	5.0.6 or higher with TCP/IP
Sun Solaris 32-bit (SPARC)	020	9 (SunOS 5.9) or higher
Sun Solaris 64-bit (SPARC)	320	9 (SunOS 5.9) or higher
Sun Solaris 64-bit (x64/x86)	420	10 (SunOS 5.10) or higher

Patches

Platform	Patch	Notes
IBM AIX 5L 5.2	xlc.rte.aix50.apr2006.ptf	This patch includes updates to the C++ runtime that are required by the version 9 compiler. You can search for the patch by name from the Support and Downloads section of the IBM web site at **http://www.ibm.com/support/us/**.
Red Hat Linux (x86)	libstdc++.so.6	This version of the standard C library is required by the version 9 compiler. If it's not already installed on your system, you can get it by installing GCC 3.4.6, which is available from **http://gcc.gnu.org/releases.html**.
SCO OpenServer	SLS OSS646C	This patch updates the UNIX Runtime System and Development System on SCO OpenServer 5.0.6 and 5.0.7. You will not be able to install Synergy/DE 9 without first installing this patch. You can download the patch from the support section of the SCO web site at **http://www.sco.com/support/**.
	N/A	If you are using SCO OpenServer 5.0.6 with the scoansi terminal type, see the technical article (TA) 112082 in the support section of the SCO web site at **http://www.sco.com/support/**.
Sun Solaris (SPARC) v9	112963-32 (all) 111711-16 (32-bit) 111712-16 (64-bit)	These patches prevent a segmentation fault that may occur when running *xf*ServerPlus. 112963 is a linker patch. 111711 (32-bit) and 111712 (64-bit) are shared library patches for the C++ runtime that are required by the version 9 compiler. Solaris patches can be downloaded from the Patches and Updates section of the Sun web site at **http://sunsolve.sun.com/pub-cgi/show.pl?target=patchpage**. To ensure you are getting the most recent revision of the patch, search on only the first six digits of the patch number. (The two digits following the hyphen are the revision.)

Platform	Patch	Notes
Sun Solaris (SPARC) v10	119963-08 118833-36	119963 is a shared library patch for the C++ runtime. 118833 is a kernel patch. Both are required by the version 9 compiler. See the information above (in the Notes column for SPARC v9) regarding downloading Solaris patches.
Sun Solaris (x86) v10	119964-08 118855-36	119964 is a shared library patch for the C++ runtime. 118855 is a kernel patch. Both are required by the version 9 compiler. See the information above (in the Notes column for SPARC v9) 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.

⚠ Run the **setsde** script (located in the synergyde directory) to enable the Synergy/DE environment for the products you have installed and licensed. You may want to add this script to your start-up file.

Setsde can be run by any UNIX shell except **csh**. We use **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 9.3 is supported on the HP OpenVMS Alpha platform and on HP OpenVMS I64 Integrity servers.

Platform	System code	Version	Patches
Alpha[a]	200	7.2 or higher	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	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] For a cluster, we recommend using version 7.3-1 or higher.

Patches

The OpenVMS patches listed in the table above are available for download from the IT resource center of the Hewlett-Packard web site at **http://www11.itrc.hp.com/service/patch/mainPage.do**. To find the most recent version of a patch on the Hewlett-Packard web site, you can browse the patch list for your version of OpenVMS or 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 web site at the URL given above. 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 9 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 version 9 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 you not use the version 9 compiler; instead, use the compiler distributed as **dbl8**. (Refer to the version 9.1 release notes, REL_DBL.TXT, 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, *xf*Server, *xf*ServerPlus, *xf*NetLink) all use TCP/IP for data transport. 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 *xf*Server. It logs each TCP/IP system call we make and its status.

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. If you use a host name, 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 3-50 for details.

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

OpenSSL Requirements

The following Synergy/DE features require OpenSSL:

- Synergy HTTP document transport API to send and receive documents via HTTPS
- Data encryption
- Client/server encryption with *xf*Server or *xf*ServerPlus

This section lists the required versions of OpenSSL for the various operating systems. For information on where to obtain OpenSSL, see Synergex KnowledgeBase article 100001979.

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:

- HTTPS, see the "Synergy HTTP Document Transport API" chapter in the *Synergy Language Reference Manual*.
- Data encryption, see "Data encryption" in the "Welcome to Synergy Language" chapter of the the *Synergy Language Reference Manual.*
- *xf*Server encryption, see "Using Encryption" on page 3-8.
- *xf*ServerPlus encryption, see "Using Encryption" in the "Configuring and Running xfServerPlus" chapter of the *Developing Distributed Synergy Applications* manual.

WIN

Synergy/DE 9.3 supports OpenSSL version 0.9.8.

UNIX

On UNIX, the supported version of OpenSSL varies by platform. The table on page 1-12 indicates the version of OpenSSL that is *expected* on the various UNIX platforms. (Note that OpenSSL versions usually include a letter, as in 0.9.7m; for our purposes, the letter does not matter.) The Synergy/DE features that use OpenSSL will look for the expected version and display an error if it is not found. Specifically, the OpenSSL shared libraries required by Synergy/DE are **libcrypto.so.#.#.#** and **libssl.so.#.#.#**, where #.#.# is the version number, e.g., 0.9.7.

If you have another version installed, and cannot install the expected version, you can do one of the following:

- Create a link with the appropriate name. For example, if on your platform Synergy is expecting OpenSSL version 0.9.7 and you have 0.9.8, you'd create a link like this:

  ```
  ln -s libcrypto.so.0.9.8 libcrypto.so.0.9.7
  ln -s libssl.s0.0.9.8 libssl.so.0.9.7
  ```

  This will cause any application looking for OpenSSL version 0.9.7 to use 0.9.8 instead.

- Run **makessl**, located in DBLDIR:csrc, to relink **synssllib.so** against your version. For example, if you have some applications that require OpenSSL version 0.9.8, but Synergy is expecting 0.9.7, you can use this method to direct only Synergy to use 0.9.7. The **makessl** file includes instructions on its use.

Note that the absolute minimum version is 0.9.7 on all platforms.

Platform	System code	Expected OpenSSL version
HP Tru64	021	0.9.7
HP-UX 32-bit (PA-RISC)	009	0.9.7
HP-UX 64-bit (PA-RISC)	309	0.9.7
HP-UX 64-bit (Itanium)	509	0.9.7
IBM AIX 5L 32-bit	004	0.9.8
IBM AIX 5L 64-bit	304	0.9.8
Linux (x86)	028	0.9.7[a]
Linux (x64)	428	0.9.7[a]
SCO OpenServer	003	0.9.7
Sun Solaris 32-bit (SPARC)	020	0.9.8
Sun Solaris 64-bit (SPARC)	320	0.9.8
Sun Solaris 64-bit (x64/x86)	420	0.9.8

[a] Linux uses its own versioning system in naming the library files; they do not use the three-digit version number shown here, but are rather named **libcrypto.so.4** and **libssl.so.4**.

VMS

Synergy/DE 9.3 supports HP SSL version 1.3, which is based on OpenSSL 0.9.7e.

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**.

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. Contact your Synergy/DE account manager for more information about this product.

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," below.
2. Install the Synergy/DE products; License Manager is included in the core components. See "Installing License Manager" on page 2-5.
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 grace period. See "Installing configuration keys" on page 2-8.

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 for network licensing.

Local licensing

Local licensing uses License Manager server (**synd**) to manage licenses on a single workstation. You can configure local licensing on any Windows system. Each workstation 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 the workstation. (You can 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 network license clients. You can configure network server licensing on any Windows machine to manage licenses across a local area network (LAN) from any Windows client (except terminal server machines). You can also configure network server licensing on Terminal Services systems to manage licenses for clients attached to the terminal server machine. When installing the license clients, you will be prompted for the name of the license server that the client will use.

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.

A workstation that runs Synergy/DE from a shared drive *must* be configured as a network license client; it cannot be configured for local licensing or as a network license server. See "Installation considerations" on page 1-3 for additional information.

When using network licensing, configuration keys need be installed only on the license server machine, not on each license client. 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.

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 is defined as a licensed workstation.) 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, and applications already running will display a message stating that usage has been exceeded. This message appears every 10 minutes until the number of concurrent users drops below the user maximum.

If a licensed Synergy/DE application on a license client becomes idle for an extended period of time, its license can be taken by another client. When the idle application becomes active again, it re-acquires a license.

When using a Windows non-server machine as a license server for other Windows non-server machines, you should be aware that Windows permits only 10 concurrent connections. Mapped drives count as connections. Though License Manager connections are quick and temporary, 10 connections may not be sufficient for a large network of clients. When using a true Windows server machine as the license server, there is no such limit.

Understanding user names and passwords

Network licensing may require additional setup to ensure that license clients can access the license server. Do one of the following:

- Configure a Windows server machine as a domain controller to provide a common source of password authentication for your license clients. In this configuration, the user name and password are set up as a domain-level account; you do not have to have accounts on both the license client and the license server. Note that the domain controller machine and the license server machine can be two separate machines.
- If you do not have a domain controller, create a user account on the license server machine that uses the same user name and password as the account on the license client machine. Note that using a domain controller greatly simplifies the set-up required for network licensing. If you have difficulty keeping the user names and passwords in sync, you may want to consider adding a domain controller to your network.

See “Troubleshooting License Manager” on page 2-14 for information on resolving access errors.

Installing License Manager

1. Follow the instructions included with your distribution to install the Synergy/DE products. License Manager is installed with the core components.
2. When prompted for the license configuration, choose either Network Server (if the machine will have local licensing or be a license server) or Network Client (if the machine will be a license client).
3. If you chose Network Server, in the Unique Licensee Name field, enter the name that you provided to Synergex when you ordered Synergy/DE products for this workstation. 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 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 chose Network Client, in the Server Name field, enter the name of the license server machine.

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 web site. For more information about Synergy/DE Developer Support and Online Keys access, contact your Synergy/DE customer service representative. See “Obtaining keys online” on page 2-6.
- **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 manually (by mail or fax)” on page 2-7.

Obtaining keys online

Obtaining keys for the current workstation

Follow these instructions to obtain configuration keys for the workstation from which you perform this procedure. The workstation must have Internet access. If you want to obtain configuration keys for several workstations at once, or for a workstation other than the one you are currently using, see "Obtaining keys for several workstations at once" below.

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

1. Start the Synergy Configuration Program (from the Start menu select Programs > SynergyDE > Utilities > Synergy Configuration Program).
2. On the Licensing tab, click the Obtain Keys Online button. This will launch your web browser and display a Resource Center login page.
3. Log in with your Resource Center user name and password, and then follow the instructions on screen to obtain configuration keys for your Synergy products.
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; you'll use it to install the keys (see "Installing configuration keys" on page 2-8).

Obtaining keys for several workstations at once

If you need to install configuration keys on several workstations, it may be more efficient to get all the keys at once in a single Synergy key file. You can also use this procedure when you need to obtain keys for a single workstation that you are not currently using. To complete this procedure, you will need your Resource Center user name and password.

1. Gather the registration information for all the workstations by doing one of the following:
   - To print the registration information from the Synergy Configuration Program, go to the Licensing tab and click the Print Registration Information button. See "Obtaining keys manually (by mail or fax)" on page 2-7 for instructions on completing the Registration Information dialog box. (Note that this option prints the *entire* license agreement, which includes the registration information.)
   - To 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.
   - To display the registration information on screen from the command prompt, run **lmu** with the **-b** option.
2. Start your web browser and go to the Synergex web site at **http://www.synergex.com**.

3. Log into the Synergy/DE 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. Use the Generate More Keys button to add additional licenses to the Synergy key file.

4. After you have requested keys for all the workstations, 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. This file will have keys for all the workstations; you can copy it to each workstation to install the keys. See "Installing configuration keys" on page 2-8.

Obtaining keys manually (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 Start menu select Programs > SynergyDE > Utilities > 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 2-8.

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

Note: The key file may contain keys for more than one system. When you perform this procedure, only the keys for the particular system are installed. To see which keys will be installed, use the View Key File button in the Synergy Configuration Program.

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. See page 2-32 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 2-26 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.

You can also install keys by running **lmu** with the **-e** option from the command line. See "The lmu Utility" on page 2-26 for more information.

Changing license configurations

Do not remove registry entries or uninstall licensing to change server to client or to alter a client's server.

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.

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 2-10.

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 name of the network license server in the Server name field. The server must have the necessary licensing in place before you can run applications on the client.

4. Click OK.

Changing a license client to use a different license server

1. Start the Synergy Configuration Program.

2. Click the Advanced button on the Licensing tab.

3. In the Advanced License Manager dialog, change the name in the Server name field. The license server must have the necessary licensing in place before you can run applications on the client.

4. Click OK.

Changing a license client to a license server

You must be a member of the administrator group to change the license configuration from a license client to a license server.

1. From the command line run **lmu -k**. The **-k** option requires a password; call Synergy/DE Developer Support to get the current password.

If you have configured License Manager to run any Synergy/DE products, after running **lmu -k** you'll need to reconfigure all products.

2. Start the Synergy Configuration Program.
3. When you are prompted to configure licensing, select server and complete the appropriate fields.

Starting and stopping License Manager server (synd)

The License Manager server, **synd**, is a background process that runs on License Manager network servers and on workstations that have local licensing. **Synd** runs as the service **synlm**. See "The synd Program" on page 2-25 for **synd** syntax and command-line options.

You must be a member of the administrator group to start or stop License Manager server.

Starting License Manager server

Starting License Manager server automatically

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 page 2-25 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 2-13.

Stopping License Manager server

Do any of the following:

- 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
  ```

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 7.5 or higher. The license clients may be running older versions of Synergy/DE.

Note the following requirements:

- You must purchase a Synergy Backup License Server license.
- On the backup server machine, Synergy/DE must be installed to run locally.
- The backup server must be a license client to the primary server.

Note the following recommendations:

- 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.
- 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. See the note on page 2-12.

Once you have the primary and backup license servers configured (see below for instructions), 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. An "active" status indicates which machine is currently being used as the license server. (Note that this dialog box will always display the names of the *configured* primary and backup servers, while the status will indicate which server is *currently being used*.)

When 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 after the primary server goes down. You must restore the primary server before 21 days have passed in order to continue serving licenses. Once the primary server is restored, the serving of licenses automatically switches back to that machine.

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 (BLSRV) on the primary server; this license key enables another machine to become a backup server to this primary server.

On the backup license server

Each time you configure a license client to be a backup license server, the BLSRV license configuration counter is reduced. You'll see a warning message on the eighth configuration. When you see this message, contact your Synergy/DE customer service representative to obtain a replacement configuration key for your BLSRV license. When you get the replacement key, configure it as described in "Installing configuration keys" on page 2-8. You can check how many configurations you have remaining and how many replacement keys have been installed; see figure 2-3 on page 2-30.

The instructions below presume you are configuring a new machine as the backup license server. If you are configuring a machine on which Synergy/DE is already installed, verify that the machine is a license client of the primary license server, and then complete steps 3 through 5.

1. Install Synergy/DE. You must install at least core components.
2. When the installation program prompts you for the license configuration, select Network Client and type the name of the primary server in the Server Name field. (Note: If the backup server machine had a previous version of Synergy/DE installed, you will not be prompted for the license configuration during installation. If necessary, you can change the license configuration to network client after installation. See "Changing license configurations" on page 2-9.)
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.

On the license clients

When you install Synergy/DE on the license clients, specify the primary server 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.

Changing the backup license server for a primary server

To change which machine is the backup license server to a particular primary license server, configure a new machine to be a backup license server following the instructions in "On the backup license server" above. The old backup server will automatically cease being a backup server.

Changing the primary server for a backup license server

Follow these instructions if you want to change the primary server machine and retain the backup license server as a backup server to a new primary.

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 one, this is all you need to do.
2. If the new primary server has a different name than the old one, do the following on the backup server:
   - Start the Synergy Configuration Program.
   - Click the Advanced button on the Licensing tab.
   - Change the value in the Server name field to the name of the new primary server.
   - Click OK.

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.

To request verbose logging,

1. Start the Synergy Configuration Program.
2. Click the Advanced button on the Licensing tab.
3. In the Advanced License Manager dialog box, select Enable verbose logging.
4. Type a logfile name, if desired. Include the complete path. If the logfile 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.
5. Click OK.

Remember to review and clear the log from time to time. Do not leave verbose logging enabled indefinitely because the disk will soon fill up.

You can also set verbose logging manually in the Windows registry by setting DEBUGLOGGING to 1 under **HKEY_LOCAL_MACHINE\SOFTWARE\Synergex\Synergy License Manager**. To specify a different location or name for the log file, set DEBUGLOGPATH to the path and filename in the registry under **HKEY_LOCAL_MACHINE\SOFTWARE\ Synergex\Synergy License Manager**.

Uninstalling License Manager

License Manager is uninstalled when you uninstall Synergy/DE. (To uninstall Synergy/DE, use the Windows Add/Remove Programs dialog box.) 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

If you reformat or replace a hard drive that contains the Windows system directory—even if that hard drive does not contain any Synergy/DE products—you'll need to reinstall your Synergy/DE products and reinitialize License Manager.

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 -r** at the command line to change the licensee name. (You cannot change the licensee name from the Synergy Configuration Program.) Note that you must be a member of the administrator group to run this command because it restarts License Manager server.

To change the licensee name, enter

```
lmu -rNewLicenseeName -ns
```

Cannot access Network License Server error

This error can occur on license clients and on machines with local licensing. This error may occur when running a Synergy application or when configuring the license client with **lmu**. The "Cannot access Network License Server" error will be accompanied by a system error number, which can help determine the source of the problem. Below, we discuss some typical system errors.

System error 2

System error 2 is a "file not found" error. It means that **synlm** could not be found on the license server machine. Assuming you have installed License Manager on the server machine, either **synlm** is not currently running or else the client is not pointing to the correct license server. If you are on a machine with local licensing, the most likely reason for this error is that **synlm** is not running; start with step 3 below.

Follow these steps to determine what the problem is:

1. On the license client, run the Synergy Configuration Program. On the Licensing tab, click the Advanced button. The Advanced License Manager dialog box will display the name of the server that the license client is attempting to use. (Note that you will likely see an error message when you run the Synergy Configuration Program. Click OK in the message dialog box, and the program will open.)
2. If the server displayed is not the correct license server for this client, enter the name of the correct server in the Server name field of the Advanced License Manager dialog box.

You must be a member of the administrator group to perform the following steps on the license server machine.

3. If the server displayed is the correct license server for this client (or you are on a machine with local licensing), 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.
4. If attempting to run the Synergy Configuration Program results in an error, go to a command prompt and run **synd -x** to unregister the service.
5. If you see a message stating that **synlm** has been removed, try running the Synergy Configuration Program again to register and start **synlm**.
6. If you see a message stating that the **synlm** service does not exist, it means that either the service was not installed or has been deleted. Run the Synergy/DE installation and re-install core components. See “Installing License Manager” on page 2-5 for more information.

System error 5 or system error 1326

System error 5 means “access denied”, and system error 1326 means “logon failure: unknown user name or bad password”. For both errors, the problem is likely that the license client does not have access rights to the license server.

- If your network includes a Windows domain controller, the license client must be logged into the domain account and the license server must be a member of that same domain or a trusted domain.
- If your network does not include a Windows domain controller, there must be a local user account on the license server machine that matches exactly the user name and password of the local account that the user used to log onto the client machine.

In either case, to verify that the client can access the license server machine, attempt to map a drive to the license server. The easiest way to do this is by browsing the network in Windows Explorer and selecting the license server machine. (Attempting to map a drive is a useful way to test your access to the license server, because License Manager server uses the Named Pipes protocol, which uses the same low-level transport mechanism as mapped drives.)

If you are prompted for a user name and password when you try to map the drive, and your network includes a Windows domain controller, this tells you that the server machine is not in the same domain as the client machine. Contact your network administrator for assistance. If your network does not include a domain controller, the problem is that the user name and password used to log onto the client are not present on the server. You can solve this problem by setting up an account on the server using the correct user name and password.

If you are not prompted for a user name and password when you map the drive, there is likely a network communication problem that is preventing your access to License Manager. Contact your network administrator for assistance.

It may not always be immediately obvious who the “user” is on the license client. For example, if you were running *xf*ServerPlus on a license client, the account used to run *xf*ServerPlus sessions would be the user for licensing purposes, and would therefore need to have permissions to open a pipe—or map a drive—to the license server. In some instances, the user account under which IIS runs is the user for licensing purposes. (See the Synergex KnowledgeBase article 100001601 for more information on this scenario.) In order to test whether, for example, the *xf*ServerPlus user can access the license server, you would need to log on as the *xf*ServerPlus user and then attempt to map a drive to the license server.

Setting Up License Manager on UNIX

License Manager is a set of utilities that controls the use of Synergy/DE products, and 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 License Manager Kit. Contact your Synergy/DE account manager for more information about License Manager Kit.

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 Language. See "Installing License Manager" below.
2. Obtain license configuration keys from Synergex. See "Obtaining configuration keys" on page 2-18.
3. Configure License Manager (i.e., install the configuration keys) to allow your Synergy/DE products to continue running after the 14-day grace period. See "Installing configuration keys" on page 2-19.

⚠ 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 run 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 2-20 and "The synd Program" on page 2-25.)

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 2-26.

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 web site. 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. You can obtain keys for a single system or for several systems at once. The configuration keys are delivered in a Synergy Key file.

1. Locate the file **register.lm** in the synergyde/lm directory and print it. If you want to obtain keys for several systems, print this file for each system.
2. Start your web browser and go to the Synergex web site at **http://www.synergex.com**.
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. If you need to install configuration keys on several systems, it may be more efficient to get all the keys at once in a single Synergy key file; use the Generate More Keys button to add additional license to the key file.

4. When prompted by the Online Key Generation program, 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 2-19).

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 -ffilename.skf
```

This command runs the **lmu** utility and installs keys for all products in the specified key file. See page 2-32 for a list of errors returned by **lmu**.

If you do not have a Synergy key file

1. Run 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 page 2-32.

   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 PSDE9.

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 (as is the case with the LMK license), 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 LMK9 -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 run **install.sde**.

For information about the **synd** command-line syntax and options, see "The synd Program" on page 2-25.

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
```

The License Manager Utility, **lmu**, enables you to specify debugging options for **synd**. See "The lmu Utility" on page 2-26 for the **lmu** syntax and options.

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 and 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 Language” chapter of the *Synergy 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.

Troubleshooting License Manager

This section includes information on several common errors.

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 Language 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 (versions 6, 7, and 8), 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 running 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 and 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.

> The MSGSEG parameter may have a different name on systems from different UNIX manufacturers.

Other errors

Synergy Language and other Synergy/DE products can generate errors that pertain to the License Manager. For individual error descriptions, see the "Error Messages" chapter of *Synergy Language Tools*.

The synd Program

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

Syntax

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

Arguments

option

One or more of the following:

WIN

h	Display a list of **synd** options.
q	Stop the License Manager server process.
r	Register the **synlm** service.
rs	Register the **synlm** service (if it is not already registered) and then start it.
v	Display the **synd** version.
x	Unregister the **synlm** service. If the service is running, **-x** will stop it and then unregister it.

UNIX

<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.
h	Display a list of **synd** options.
p	Ping the License Manager daemon to see if it is responding. This option also displays the version of **synd** that is running.
q	Stop the License Manager daemon. You must be UID=0 to do this.
v	Display the **synd** version.

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.

Syntax

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

Arguments

option

One or more of the following:

<none> Display concurrent usage and configuration information. See figure 2-1 on page 2-29. On Windows, 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 2-31 for sample output.

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

b Brief display of license information. On UNIX and on a Windows license server configuration, includes licensee name, registration string, and system code. On a Windows license client configuration, includes license server name, registration string, and system code.

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.

h Display a list of **lmu** options.

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

k *password* Reinitialize License Manager with a *password* obtained from Synergy/DE Developer Support. This password is valid for seven days. On Windows, you must use **lmu -d** to complete reinitialization. On UNIX, after using **lmu -k**, you must use **install_lm** to complete reinitialization; 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 option may not display all data in the file; it displays data only for the current machine. 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.

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. Specify **-p** with **-u** when the producer ID is other than 999. See "Displaying license usage information" on page 2-31 for sample output.

v — Display current version.

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

WIN

c*name* — Initialize License Manager for licensee or server *name.* Used with **-nc** or **-ns**. If *name* contains spaces, it must be enclosed in quotation marks (" "). The minimum number of characters for licensee *name* is 4; the maximum is 50. For server *name*, the minimum is determined by your Windows operating system; the maximum is 15.

- To initialize License Manager on a license server for licensee *name*, use **lmu -c***name* **-ns**.
- To initialize License Manager as a license client to server *name*, use **lmu -c***name* **-nc**.

d — Initialize License Manager (prompted).

e — Configure product (prompted).

nb — Initiate the installation of a backup server. This option cannot be used with other options. Licensing must be configured on the primary server before running this option on the backup server. 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**.

r*name* Reinitialize License Manager. Used with **-nc** or **-ns**. Use this option to change a license client to use a different license server, change the licensee name for a license server prior to configuring products, or change a license server to a license client.

- To change a license client to be a client of a different license server, use **-r***new_server_name* **-nc**.
- To change the licensee name on a license server prior to configuring products, use **-r***name* **-ns**.
- To change a license server to be a license client of server *name*, use **-r***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 license server; you must use **lmu -k**. See "Changing a license client to a license server" on page 2-10 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 the DEBUGLOGGING option, because you can match up the ID with the usage or debug output. On Terminal Services, this option also displays details such as whether the workstation is a client or console and whether install mode is enabled.

UNIX

c*name* Initialize License Manager for licensee *name*. If *name* contains spaces, it must be enclosed in quotation marks (""). The minimum number of characters for *name* is 4; the maximum is 50.

q Display Synergy/DE message queue information.

r*name* Reinitialize License Manager for licensee *name* prior to configuring products.

1 Stop debug logging and send a reset signal to License Manager. See "Error logging" on page 2-21 for more information.

2 Start debug logging by License Manager. See "Error logging" on page 2-21 for more information.

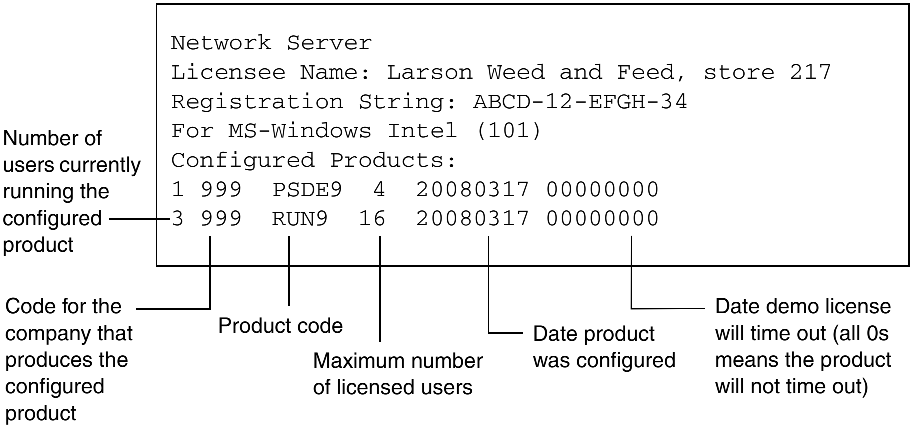
Discussion

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

For a list of error messages returned by the **lmu** utility, see the table on page 2-32.

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, current number of users, and maximum number of licensed users. On a Windows license client, you'll see the products that are configured on the license server. On a Windows workstation with local licensing, the number of concurrent users displays "W" (for workstation). Figure 2-1 shows a sample report on a Windows license server. Note that you can run **lmu** with the **-o** option to send this information to a file.

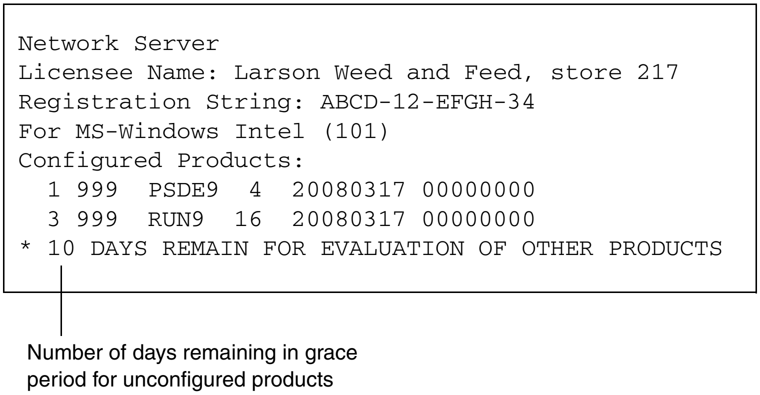


```
Network Server
Licensee Name: Larson Weed and Feed, store 217
Registration String: ABCD-12-EFGH-34
For MS-Windows Intel (101)
Configured Products:
1 999  PSDE9   4  20080317 00000000
3 999  RUN9   16  20080317 00000000
```

Number of users currently running the configured product
Code for the company that produces the configured product
Product code
Maximum number of licensed users
Date product was configured
Date demo license will time out (all 0s means the product will not time out)

Figure 2-1. Sample lmu report for configured products.

If the 14-day grace 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 below.

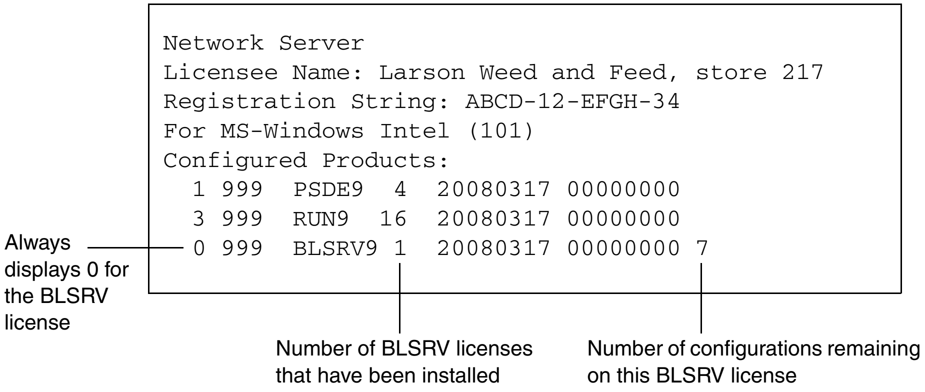


```
Network Server
Licensee Name: Larson Weed and Feed, store 217
Registration String: ABCD-12-EFGH-34
For MS-Windows Intel (101)
Configured Products:
  1 999  PSDE9  4  20080317 00000000
  3 999  RUN9  16  20080317 00000000
* 10 DAYS REMAIN FOR EVALUATION OF OTHER PRODUCTS
```

Number of days remaining in grace period for unconfigured products

Figure 2-2. Sample lmu report with grace period remaining.

If you are using a backup license server (Windows only), the **lmu** report will tell you how many BLSRV keys have been installed and the number of configurations remaining as shown in figure 2-3 below. When the latter reaches 1, you should contact your Synergy/DE customer service representative to obtain a replacement configuration key for your BLSRV license.



```
Network Server
Licensee Name: Larson Weed and Feed, store 217
Registration String: ABCD-12-EFGH-34
For MS-Windows Intel (101)
Configured Products:
  1 999  PSDE9  4  20080317 00000000
  3 999  RUN9  16  20080317 00000000
  0 999  BLSRV9 1  20080317 00000000 7
```

Always displays 0 for the BLSRV license

Number of BLSRV licenses that have been installed

Number of configurations remaining on this BLSRV license

Figure 2-3. Sample lmu report showing backup license server information.

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 (RUN9) and Workbench (PSW9). Each product is configured for 10 users. There are currently two users using runtime licenses and one user using a Workbench license.

```
C:\lmu RUN9 PSW9
License: RUN9 Producer: 999    (10 User)
  Configured on August 20, 2009
  Current usage: 2

License: PSW9 Producer: 999    (10 User)
  Configured on August 20, 2009
  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 RUN9 PSW9
License: RUN9 Producer: 999    (10 User)
  Configured on August 20, 2009
  Current usage: 2
      Seat 1: MICKEYf75dc5f1 - 1 active process
      Seat 2: BARNEYbe71aa31 - 1 active process (Non-interactive)

License: PSW9 Producer: 999    (10 User)
  Configured on August 20, 2009
  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
        PSDE9            Professional Series Development Environment
        RUN9             Synergy Runtime
IJKL-67-MNOP-93 - Larson Weed and Feed, store 354
        RUN9             Synergy Runtime
        PSW9             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:
      PSDE9      Professional Series Development Environment
      RUN9       Synergy Runtime
```

Usage

- Windows. This example shows how to reinitialize a network license client for server *Tiger* (i.e., it can be used to change the server name). This syntax is also used to change a license server to a license client.

  ```
  lmu -r Tiger -nc
  ```

- UNIX. This example shows how to configure a license for the runtime (RUN9) from the command line. This command includes the producer ID (999) and the configuration key.

  ```
  lmu -a RUN9 -p 999 -i 58W4-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 from 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 2-9. (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)
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. See "Troubleshooting License Manager" on page 2-14. (Windows)

lmu Error Messages (Continued)

Message	Cause
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)
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. The licensee name can only be changed prior to configuring products.
Configuration limit for BLSRV exceeded	You have exceeded the limit for configuring the backup license server. You can configure a backup license server only eight times for a given BLSRV 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. For pre-version 7.5, this message could also mean that the version of **lmu** is incompatible with the license file and should be updated. (Windows)
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 2-22. (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.)

lmu Error Messages (Continued)	
Message	**Cause**
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)
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. The licensee name can only be changed prior to configuring products.
Configuration limit for BLSRV exceeded	You have exceeded the limit for configuring the backup license server. You can configure a backup license server only eight times for a given BLSRV 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. For pre-version 7.5, this message could also mean that the version of **lmu** is incompatible with the license file and should be updated. (Windows)
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 2-22. (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.)

lmu Error Messages (Continued)	
Message	**Cause**
Invalid language code	You entered an invalid language code. See page 2-27 for the valid codes.
Invalid registration string	The license file is not registered for this machine. Contact Synergy/DE Developer Support.
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 workstation as a license server. To be a license server, the workstation must have Synergy/DE installed to run from the local hard drive. (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 15-character limit. (Windows)
No Synergy Key Files specified	You ran **lmu** with the **-f**, **-m**, or **-t** option but didn't supply a Synergy key file name.

lmu Error Messages (Continued)	
Message	**Cause**
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 BLSRV, have been configured on the primary license server. (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 you are entering the configuration key that 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 *xf*Server

Getting Started with *xf*Server

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 *xf*Server and how to use it.

With the exception of the **rsynd** program syntax, this chapter describes *xf*Server. For information on *xf*ServerPlus, see *Developing Distributed Synergy Applications: Using xfNetLink and xfServerPlus*.

There are four steps that you should take in preparation for using *xf*Server:

1. **Install your client and server systems**
   - On your server system, follow the instructions that came with your Synergy/DE distribution to install *xf*Server.
   - On your client systems, install Synergy Language 6.1 or higher. (Version 9.3 or higher is required for encryption.) Synergy Language includes client capabilities in the runtime.
   - Review the Synergy Language and *xf*Server 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 3-50.

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 3-5 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 Encryption" on page 3-8.

Restrictions

Note the restrictions for the following statements and subroutines when used with *xf*Server.

- COPY subroutine—Not implemented on OpenVMS *xf*Server.
- GETFA subroutine—Only the following keywords can be used with *xf*Server: 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 *xf*Server 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 *xf*Server; however, if you choose this configuration, the printer must be attached directly to the server machine. Network printers are not supported from *xf*Server on Windows.
- MERGE statement—Not implemented on OpenVMS *xf*Server.
- 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 *xf*Server.
- 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); if they are not set, it is placed in C:\Documents and Settings\All Users\Shared Documents (XP/2003) or C:\Users\Public\Public Documents (all other Windows platforms).
    - On UNIX, it is placed in /tmp.
    - On OpenVMS, it is placed in SYS$SCRATCH.

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

Why use environment variables with *xf*Server?

Although you can hard-code server specifications in your program, to keep your *xf*Server 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 UNIX and OpenVMS *xf*Server, to point to the user's home directory, use

  `DAT=@`*server_name*

  On Windows, this will point to the default directory, as defined by RSFILPATH set on the Windows server. (See RSFILPATH in the "Environment Variables" chapter of *Environment Variables and System Options* for more information about this environment variable.)

- To point to the SRV environment variable, as defined by *xf*Server, 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 *xf*Server machine. For instructions on defining environment variables for the *xf*Server service, refer to the following operating system-specific information:

  - Windows, see "Defining environment variables for xfServer services" on page 3-19
  - UNIX, see "Defining environment variables for xfServer processes" on page 3-25
  - OpenVMS, see "Defining logical names for xfServer processes" on page 3-31

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.

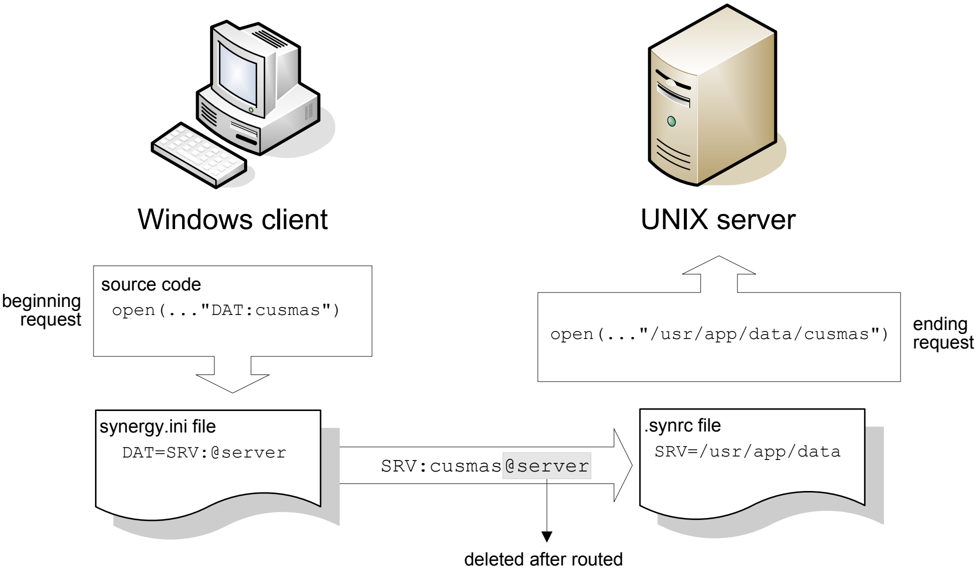


Figure 3-1. Translating a request from a Windows client to a UNIX server using an environment variable.

Prefetching Records

You can improve sequential READS performance with the prefetch feature. This feature enables the client to prefetch sequential records from the server and store them in a buffer until they are needed. You can use this feature with files of any type that are open for input, with relative files that are open for update, or with ISAM files that are open for update and use the LOCK:Q_NO_TLOCK option.

By default, prefetching is off. To turn prefetching on, set the SCSPREFETCH environment variable 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. See SCSPREFETCH in the "Environment Variables" chapter of *Environment Variables and System Options* for more information on setting this environment variable.

If you currently have Q_NO_LOCK set and wish to use prefetching, you can set system option #55. This will automatically map 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.

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

Using Encryption

The *xf*Server encryption feature enables you to encrypt the transfer of sensitive data across a network. *xf*Server 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. 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", below.

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 and the cipher (i.e., level of encryption) to use. The three cipher options (high, medium, low) map to specific cipher suites, which may vary by OpenSSL version. See "Setting up the server and client machines for encryption" on page 3-9.

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 and System Options* for more information.)

Specifying the data to encrypt for slave encryption

When *xf*Server 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.)

  See OPTIONS in the "Synergy Language Statements" chapter of the *Synergy Language Reference Manual* for more information about using the **/encrypt** option with the OPEN statement.

WIN, UNIX

- 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.

  The advantage to using the network encryption flag is that 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 have to add the **/encrypt** option to every OPEN statement that opens the file.

  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.

  > The network encryption flag applies to *all* remote access. 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, Functions, and Classes” chapter of the *Synergy Language Reference Manual* and isutl in the “Synergy DBMS” chapter of *Synergy Language Tools*.

Setting up the server and client machines for encryption

Follow these steps to set up your *xf*Server system to use encryption. For details on which version of OpenSSL is required for your operating system, see “OpenSSL Requirements” on page 1-11.

1. Install OpenSSL on your server machine. See Synergex KnowledgeBase article 100001979 for the latest information on where to obtain OpenSSL for the various OSs and details on installation.

   - For Windows, go to **http://www.openssl.org/related/binaries.html** and follow the link to download the version for your platform.
   - For UNIX, obtain the OpenSSL security kit specific to your operating system and version from your operating system vendor, if one is available. For some platforms, you may have to download and build the shared libraries. See Synergy KnowledgeBase article 100001979 for details.
   - For OpenVMS, use the operating system-supplied libraries, HP SSL, available from **http://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, use the Synergy Configuration Program to set the OPENSSLLIB environment variable to the location of the OpenSSL libraries. If the files are on a remote machine, use a UNC path rather than a mapped drive to specify the location. If OPENSSLLIB is not set or the OpenSSL files are not found there, **rsynd** uses PATH to locate the OpenSSL libraries.

      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 most UNIX platforms, **rsynd** looks in the /usr/lib directory for the libraries. If they are not found there, it checks /lib. On the following platforms, the specified directory is checked first, followed by /usr/lib and /lib:

        - HP-UX 64-bit PA-RISC (309): /usr/lib/pa20_64
        - HP-UX 64-bit Itanium (509): /usr/lib/hpux64
        - IBM AIX 64-bit (304): /usr/lib64
        - Sun Solaris 64-bit SPARC (320): /usr/lib/64
        - Sun Solaris 64-bit i386 (420): /usr/lib/amd64
        - Linux 64-bit (428): /lib64

      If the libraries cannot be found, 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 "Making a certificate request" in the "Synergy HTTP Document Transport API" chapter of the *Synergy 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 *xf*Server 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, if OpenSSL is installed in the default location (Windows\System32 directory), you don't need to set anything. If you install to another location, set OPENSSLIB to the location of the libraries. If OPENSSLLIB is not set, PATH is used to find the OpenSSL libraries.
    - On UNIX clients, the library path is used to find the OpenSSL libraries. This is LIBPATH on AIX, SHLIB_PATH on HP-UX, and LD_LIBRARY_PATH on all other systems. For 64-bit systems, you can also use the server paths listed in step 2 above.
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 and cipher if desired. See "The rsynd Program" on page 3-39 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 certificate file and the cipher. See "Modifying the SynSrv xfServer service" on page 3-12.

> If you are running *xf*Server 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 being used, use the SLC (Socket Level Cipher) option to the GETFA routine; it returns a string identifying the cipher, e.g., "AES256-SHA", or a blank string if the channel is not encrypted. See GETFA in the "System-Supplied Subroutines, Functions, and Classes" chapter of the *Synergy 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, Functions, and Classes" chapter of the *Synergy Language Reference Manual*.

Using *xf*Server on Windows

During installation, *xf*Server is installed with default settings for port, logging, security, and encryption. The installation program also registers *xf*Server in the Windows registry using the default service name, **SynSrv**, and then starts it.

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

When you're running *xf*Server, you can use the Monitor utility (**synxfmon.exe**) to find out which files are open, who opened them, and whether those files are locked. See "The Monitor Utility for Windows" in the "General Utilities" chapter of *Synergy Language Tools*.

Modifying the SynSrv *xf*Server service

Before modifying any *xf*Server 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.

You must be a member of the administrator group to modify settings and to register and start an *xf*Server 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 Start menu, select Programs > SynergyDE > Utilities > 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 *xf*Server 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 *xf*Server to run on. Valid ports are in the range 1024 through 65535. The default is 2330.

   This option sets SCSPORT in the registry under **HKEY_LOCAL_MACHINE\SOFTWARE\Synergex\Synergy xfServer\***service_name***\Default**. If you use a non-default port, you must also use SCSPORT to specify it on the clients. See SCSPORT in the "Environment Variables" chapter of *Environment Variables and 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 *xf*Server 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.

  Selecting Secure sets SECUREMODE to 1 in the registry under **HKEY_LOCAL_MACHINE\SOFTWARE\Synergex\Synergy xfServer\***service_name***\Default.**

- **Restricted.** Restricted mode uses Windows authentication exclusively; consequently, only Windows clients are allowed. If RUSER is set on the client, it is ignored.

  Selecting Restricted sets SECUREMODE to 2 in the registry under **HKEY_LOCAL_MACHINE\SOFTWARE\Synergex\Synergy xfServer\***service_name***\Default.**

- **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.

  Selecting Non-secure sets SECUREMODE to 0 in the registry under **HKEY_LOCAL_MACHINE\ SOFTWARE\Synergex\Synergy xfServer\***service_name***\Default.**

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 *xf*Server. This account can be either a local account or a domain controller account; it cannot be a member of the administrator 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 *xf*Server.

Compress data packets. Indicate whether you want to compress data records sent between *xf*Server and its clients. Compression can significantly improve performance on low speed or busy networks, especially WANs.

Selecting this option sets SCSCOMPR to 1 in the registry under **HKEY_LOCAL_MACHINE\SOFTWARE\Synergex\Synergy xfServer\***service_name***\Default**.

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 and System Options*.

Enable encryption. Select this option to enable encryption of data between client and server, and then select the type of encryption desired, Master or Slave. See "Using Encryption" on page 3-8 for details on this feature. Selecting this option sets SSLEncryption to Master or Slave in the registry under **HKEY_LOCAL_MACHINE\SOFTWARE\Synergex\Synergy xfServer\***service_name***\Default**.

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. Selecting this option sets SSLCertificate to the filename in the registry under **HKEY_LOCAL_MACHINE\SOFTWARE\Synergex\Synergy xfServer\***service_name***\Default**.

Cipher type. Select the level of encryption desired. Selecting this option sets SSLCipherSuite to Low, Medium, or High in the registry under **HKEY_LOCAL_MACHINE\SOFTWARE\Synergex\Synergy xfServer\***service_name***\Default**. The three cipher options map to specific cipher suites, which may vary depending on OpenSSL version. You can use the SLC option to the GETFA routine to see which cipher suite is being used. See GETFA in the "System-Supplied Subroutines, Functions, and Classes" chapter of the *Synergy Language Reference Manual*.

Event 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.

Selecting this option sets ENABLELOGGING to 1 in the registry under **HKEY_LOCAL_MACHINE\SOFTWARE\Synergex\Synergy xfServer\***service_name***\Default**.

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.

Selecting this option sets ENABLELOGGING to 2 in the registry under **HKEY_LOCAL_MACHINE\SOFTWARE\Synergex\Synergy xfServer\***service_name***\Default**

4. If you want to modify environment variables for *xf*Server, click the Environment Settings button and see "Defining environment variables for xfServer services" on page 3-19 for instructions.

> 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 and System Options*.

5. Click OK in the *xf*Server 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 table on page 3-22 for status and error codes that may occur.

Adding a new *xf*Server service

You can run multiple *xf*Server services; each service must have a different port, service name, and display name. You must be a member of the administrator group to register and start an *xf*Server 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. Complete the other fields. See step 3 on page 3-12 for detailed field information.
5. If you want to modify environment variables for *xf*Server, click the Environment Settings button and see "Defining environment variables for xfServer services" on page 3-19 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 3-22.

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 *xf*Server and *xf*ServerPlus settings and for environment variables used by *xf*Server and *xf*ServerPlus.

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 *xf*Server services. Existing services are not affected.
  - The type of data access—Secure, Restricted, or Non-secure
  - The Compress data packets option
  - The Enable logging and Verbose logging options
- The following settings are applied to all newly created *xf*Server and *xf*ServerPlus services. Existing services are not affected.
  - The Enable encryption option, the Certificate file value, and the Cipher type
- Environment variables are applied to all existing *xf*Server and *xf*ServerPlus services, as well as to any new services that you create. See "Defining environment variables for xfServer services" on page 3-19 for more information on environment variables.

Understanding *xf*Server security

On Windows, *xf*Server 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 *xf*Server for authentication. The user name will be used to generate the persona used during *xf*Server 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 *xf*Server client.

Windows authentication security uses the Windows operating system to authenticate Windows clients. No credentials are passed from the client to *xf*Server. 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 *xf*Server 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 *xf*Server 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 *xf*Server will be denied.

On a UNIX client, if RUSER is not set, an error will be generated and access to *xf*Server 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 3-57 and RUSER in the "Environment Variables" chapter of *Environment Variables and System Options*.)

To use Windows authentication, the client user must have an account on a Windows domain, and the *xf*Server 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 *xf*Server 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 3-39 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 *xf*Server 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 *xf*Server 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 *xf*Server from the command line, specify the **-sspi** option when you register the service. (See "The rsynd Program" on page 3-39 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 *xf*Server. This account can be on the *xf*Server machine (a local account), or it can be a domain controller account. The account cannot be a member of the administrator group.

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

The account that you create for use with *xf*Server must allow a log-in on the machine that *xf*Server is running on. If you cannot log on to the *xf*Server 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 *xf*Server 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” checkbox, and then enter the user name and password for the account you created for use with *xf*Server. (See “Modifying the SynSrv xfServer service” on page 3-12 for details.)

If you are starting *xf*Server 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 *xf*Server account, when you register the service. (See “The rsynd Program” on page 3-39 for details on command-line options.)

Defining environment variables for *xf*Server services

See “Why use environment variables with xfServer?” on page 3-5 for a general discussion of the benefits of using environment variables with *xf*Server. If your client application includes environment variables in the format “LOGICAL:@server_name”, you must define those environment variables on the server, where *xf*Server can find them.

On Windows, the recommended method for defining environment variables for *xf*Server is to use the Synergy Configuration Program, which writes the settings to the Windows registry. (*xf*Server 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 *xf*Server services or for a specific instance of *xf*Server. Environment variables can also be defined for a specific user, but you must do so manually; see “Defining environment variables manually” on page 3-21.

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.

Defining environment variables with the Synergy Configuration Program

Defining environment variables for all services

Environment variables set in this manner will apply to all instances of *xf*Server and *xf*ServerPlus, both existing and newly created. These environment variables are set in the registry under **HKEY_LOCAL_MACHINE\SOFTWARE\Synergex\Synergy xfServer\Synrc**.

If an environment variable is already set for a specific service (see “Defining environment variables for a specific service” on page 3-20), 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 *xf*Server and will override settings made for all services. They are written to the registry under **HKEY_LOCAL_MACHINE\SOFTWARE\Synergex\Synergy xfServer\***service_name***\Synrc.**

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 manually

The recommended method for setting environment variables is to use the Synergy Configuration Program, as described above. It writes the settings to the Windows registry. If necessary, though, such as when you need to create settings for a specific user, you can edit the registry manually by following the instructions in this section.

Defining environment variables for a specific user

Environment variables defined for a specific user will override environment variables set for all *xf*Server services and those set for a specific *xf*Server 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 *xf*Server starts.

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 *xf*Server. 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.

Defining environment variables for all services or a specific service

1. Run **regedit**.
2. Navigate to the location in the registry where you want to set the environment variable:
   - To set an environment variable for all instances of *xf*Server and *xf*ServerPlus, go to **HKEY_LOCAL_MACHINE\SOFTWARE\Synergex\Synergy xfServer\Synrc**.
   - To set an environment variable for a specific instance of *xf*Server, go to **HKEY_LOCAL_MACHINE\SOFTWARE\Synergex\Synergy xfServer\***service_name***\Synrc**. This setting overrides the above setting.
3. Add a new environment variable as described in step 4 above.
4. Stop and restart *xf*Server. See "Stopping xfServer" on page 3-23 and "Starting xfServer" on page 3-22.

Starting *xf*Server

The default service **SynSrv** is started automatically when you install. However, there may be times when you need to start it manually. You must be a member of the administrator group to start *xf*Server.

To start *xf*Server 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 *xf*Server), and select Action > Start.
- At a command prompt enter

  ```
  net start servicename
  ```

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

The following codes may occur when starting or stopping *xf*Server. 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.

Code	What it means
2	You must belong to the administrator group to register or unregister the service.
3	The specified account does not exist on the machine.
5	User not found on local machine.
6	The username specified for the default user account for *xf*Server or for running *xf*ServerPlus sessions belongs to the administrator group. 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.

Code	What it means
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 *xf*Server

Before stopping or removing (unregistering) a service, you should verify that no users are connected. You must be a member of the administrator group to stop *xf*Server.

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

To stop *xf*Server, 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 *xf*Server), and then click the Stop button.
- At a command prompt enter

  ```
  net stop servicename
  ```

  where *servicename* is the name of the *xf*Server service to be stopped. The default service name is **SynSrv**.
- At a command prompt enter

  ```
  rsynd -q
  ```

  This stops the default *xf*Server, **SynSrv**, on the default port, 2330. If you are running *xf*Server with a different service name, use the **-c** option to specify the service name. If you are running *xf*Server on a non-default port, use the **-p** option to specify the port number.

To both stop *and unregister xf*Server, 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 *xf*Server, **SynSrv**, on the default port, 2330. If you are running *xf*Server with a different service name, use the **-c** option to specify the service name. If you are running *xf*Server on a non-default port, use the **-p** option to specify the port number.

Using *xf*Server on UNIX

Defining environment variables for *xf*Server processes

See “Why use environment variables with xfServer?” on page 3-5 for a general discussion on the benefits of using environment variables with *xf*Server. If your client application includes environment variables in the format LOGICAL:@*server_name*, you must define those environment variables on the server, where *xf*Server 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 *xf*Server by port number. To do this, name the file **synrc.####**, where #### is the port number, and put the file in the /etc directory. A **synrc** file for a specific port number takes precedence over the generic **synrc** file.

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 *xf*Server 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 *xf*Server. This user name will be used to generate the persona used during *xf*Server 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 *xf*Server by specifying the port number as the file extension. (A sample **synrc** file is included with your Synergy Language distribution.)
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.) *xf*Server (**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 *xf*Server

Modes

On UNIX, *xf*Server 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 *xf*Server 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 *xf*Server 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 *xf*Server access.

To run *xf*Server 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 3-57 and RUSER in the "Environment Variables" chapter of *Environment Variables and 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 *xf*Server is denied. If authentication succeeds, the user name sent to the server is used to generate the persona used during *xf*Server access.

To run *xf*Server 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 *xf*Server client.

For more information on setting RUSER, see "The setruser Utility" on page 3-57 and RUSER in the "Environment Variables" chapter of *Environment Variables and 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 *xf*Server is denied. If the user does have an account on the server, it is used to generate the persona used during *xf*Server access.

To run *xf*Server 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 *xf*Server. **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 *xf*Server 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 3-57.

- 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 *xf*Server. 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, *xf*Server 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 3-25. For more information, see RSYNDLOG and RSLOGMAX in the "Environment Variables" chapter of *Environment Variables and System Options*.

Ports

The default port used by *xf*Server is 2330. If you are running multiple instances of *xf*Server 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 a Windows client, set SCSPORT in the registry.) For more information, see SCSPORT in the "Environment Variables" chapter of *Environment Variables and System Options*. See page 3-39 for **rsynd** syntax and options.

Compression

Set the SCSCOMPR environment variable to a non-zero value in the **synrc** file on the server (see page 3-25) to compress data records sent between *xf*Server and its clients. 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 and System Options*.

Starting *xf*Server

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

1. To enable the *xf*Server daemon to grant adequate resources to clients, verify that the **rsynd** program is owned by root. In the Synergy Language 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 *xf*Server 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 *xf*Server

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

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

  ```
  rsynd -q
  ```

  This stops *xf*Server 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 *xf*Server daemon *and kill all existing connections*, use the **-c** option with **-q**:

  ```
  rsynd -q -c
  ```

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

  This stops *xf*Server 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.

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 *xf*Server on OpenVMS

Defining logical names for *xf*Server processes

See "Why use environment variables with xfServer?" on page 3-5 for a general discussion on the benefits of using logicals with *xf*Server. If your client application includes environment variables in the format LOGICAL:@*server_name*, you must define those environment variables on the server, where *xf*Server 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 page 3-34 for an example showing how two of the methods can be used together.) The information in this section applies to both *xf*Server and *xf*ServerPlus.

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 *xf*Server 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 *xf*Server 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 *xf*Server 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 *xf*Server-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 *xf*Server to ignore the **SYNRC.COM** file in the home directory of the user when setting up logicals. See "The rsynd Program" on page 3-39 for complete information on **rsynd** qualifiers.

Defining logicals for use by all instances of *xf*Server

If you have several instances of *xf*Server 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 *xf*Server will use the same logicals because this method does not enable you to specify a port. If you have multiple instances of *xf*Server, 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 *xf*Server to ignore the **SYNRC.COM** file in the home directory of the user when setting up logicals. See "The rsynd Program" on page 3-39 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 *xf*Server, which enables you to define logicals differently for separate instances of *xf*Server. **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 *xf*Server-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 *xf*Server 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 *xf*Server to ignore the **SYNRC.COM** file in the home directory of the user when setting up logicals. See “The rsynd Program” on page 3-39 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 *xf*Server 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 *xf*ServerPlus because it uses a single account to run all *xf*ServerPlus 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 *xf*Server 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 *xf*Server

Modes

On OpenVMS, *xf*Server 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 *xf*Server. 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 *xf*Server is denied. If authentication succeeds, the user name is used to generate the persona used during *xf*Server access.

To start *xf*Server 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 3-57 and RUSER in the "Environment Variables" chapter of *Environment Variables and System Options*.

Even though they are encoded, RUSER credentials should be kept confidential because they can be used with any *xf*Server 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 *xf*Server is denied. If the user does have an account on the server, it is used to generate the persona used during *xf*Server access.

To start *xf*Server 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 *xf*Server. **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 *xf*Server in non-secure mode with encryption enabled, you *must* specify a default user account.

To start *xf*Server 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 *xf*Server. 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 logfile, 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 logfile, 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.

Ports

The default port for *xf*Server is 2330. If you are running multiple instances of *xf*Server on the same machine, each will need to run on a different port. To start *xf*Server 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 *xf*Server. (For a Windows client, set SCSPORT in the registry.) For more information, see SCSPORT in the "Environment Variables" chapter of *Environment Variables and System Options*.

Compression

Set the SCSCOMPR logical to a non-zero value in the **DBLDIR:SYNRC.COM** or **SERVER_INIT.COM** file (see page 3-33) to compress data records sent between *xf*Server and its clients. 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 and System Options*.

Starting *xf*Server

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

To start *xf*Server manually, execute the following command with the desired switches while logged into the SYSTEM account. (See page 3-39 for the complete **rsynd** syntax.)

```
$ RSYND
```

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

```
$ @SYNERGY_STARTUP XFRESTART
```

*xf*Server can also be started from the **servstat** program. See "The servstat Program" in the "General Utilities" chapter of *Synergy Language Tools* for more information.

On OpenVMS, *xf*Server sets the current working directory to the user's default directory and creates a SYS$SCRATCH logical to reference the user's directory.

Stopping *xf*Server

There are several ways to stop *xf*Server (**rsynd**):

- To stop *xf*Server 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 *xf*Server *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 *xf*Server. See "The servstat Program" in the "General Utilities" chapter of *Synergy Language Tools* for more information.

The rsynd Program

The **rsynd** program is used by both *xf*Server and *xf*ServerPlus. (Using the **-w** option indicates that commands apply to *xf*ServerPlus; for additional information about *xf*ServerPlus, see the *Developing Distributed Synergy Applications* manual.) On Windows, *xf*Server runs as the **rsynd** service **SynSrv** by default, and *xf*ServerPlus runs as **xfspl** by default. On UNIX, *xf*Server and *xf*ServerPlus run as the **rsynd** daemon. On OpenVMS, *xf*Server and *xf*ServerPlus run as the **rsynd** detached process.

On Windows, do not attempt to issue **rsynd** commands from the command line while the Synergy Configuration Program is running.

Syntax

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

Arguments

option

One or more of the following:

WIN

-a

Allow persona that is a member of the administrator group as the username/password for the default user account when running *xf*Server in non-secure mode or as the username/password under which to run *xf*ServerPlus 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 *xf*Server; used with **-r** (or **-rs**) and **-w** for *xf*ServerPlus.

-c*name*

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

To run multiple instances of *xf*Server or *xf*ServerPlus, each must have a different service name, display name, and port number. See the Usage section on page 3-42.

-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** and **-encrypt**.

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

-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 *xf*Server is **Synergy/DE xfServer**; for *xf*ServerPlus, it is **Synergy/DE xfServerPlus**. Used with **-r** or **-rs**.

-encrypt*[=level]* Enable data packet encryption for *xf*Server or—if **-w** is specified—for *xf*ServerPlus. Valid values for *level* are MASTER and SLAVE. If *level* is not specified, slave encryption is used by default. Used with **-cert** and **-cipher**. See "Using Encryption" on page 3-8 for details on *xf*Server encryption or "Using Encryption" in the "Configuring and Running *xf*ServerPlus" chapter of the *Developing Distributed Synergy Applications* manual for details on *xf*ServerPlus encryption.

-h Display a list of **rsynd** options.

-n Run *xf*Server in non-secure mode. Used with **-r** or **-rs**. Does not apply to *xf*ServerPlus. 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 *xf*Server or *xf*ServerPlus listens. The default port is 2330 for *xf*Server and 2356 for *xf*ServerPlus. Used with **-q**, **-r**, **-rs**, or **-x**. *Port* updates the SCSPORT registry entry for the service.

-q Stop the *xf*Server or *xf*ServerPlus 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]* Enable remote debugging via Telnet for *xf*ServerPlus on the specified port. Used with **-r** (or **-rs**) and **-w**. *xf*ServerPlus will listen on the specified port for the Telnet client; this port number must be different than the port that *xf*ServerPlus 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 *xf*NetLink–*xf*ServerPlus connection has been made. The default is 100 seconds. See the Usage section on page 3-42.

-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 Run *xf*Server in secure mode. This is the default. Used with **-r** or **-rs**. Does not apply to *xf*ServerPlus.

-sspi Run *xf*Server in restricted security mode (Windows authentication only). See "Understanding xfServer security" on page 3-16 for more information. Used with **-r** or **-rs**. Does not apply to *xf*ServerPlus.

-u*persona* *xf*Server: Specify *persona* as the default user name and password when running *xf*Server 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.

*xf*ServerPlus: Specify *persona* as the user name and password under which to run *xf*ServerPlus 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 *xf*ServerPlus.

-v Display the current version of **rsynd**.

-w Specify that this **rsynd** request is for *xf*ServerPlus (not *xf*Server). Used with **-q**, **-r**, **-rs**, or **-x**. See the *Developing Distributed Synergy Applications* manual for information on *xf*ServerPlus.

-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.

Usage

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

The example below shows how to register **rsynd** for *xf*Server 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 *xf*Server. This *xf*Server 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"
```

The following example shows how to register **rsynd** for *xf*Server 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 *xf*Server. The default port (2330), service name (SynSrv), and display name (Synergy/DE xfServer) will be used.

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

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

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

The following example shows how to register **rsynd** for *xf*ServerPlus 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. *xf*ServerPlus 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 *Developing Distributed Synergy Applications* manual.

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

The following example shows how to register **rsynd** for *xf*Server and enable slave encryption using the default cipher and default **.pem** filename. The service will run in secure mode and use the default service name and display name.

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

The following example shows how to register **rsynd** for *xf*Server and enable master encryption. We set the cipher to high and specify a **.pem** file that has a non-default name. The service will run in secure mode and use the default service name.

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

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 *xf*ServerPlus and enable slave encryption using the default **.pem** filename:

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

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 3-22 for details. If you prefer, you can use the **-rs** option to register and start **rsynd** in one step.

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

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

UNIX

<none> Start *xf*Server. If a mode is not specified with **-n** or **-s**, *xf*Server runs in trusted mode. See page 3-26 for more information about modes.

-a Allow super user as the persona of the default user account or as the client for *xf*Server, or allow super user as the persona under which to run *xf*ServerPlus 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 *xf*Server; used with **-w** for *xf*ServerPlus.

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

See also "Stopping xfServer" on page 3-29. For more information on stopping *xf*ServerPlus, see "Running xfServerPlus on UNIX" in the "Configuring and Running xfServerPlus" chapter of the *Developing Distributed Synergy Applications* manual.

-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** and **-encrypt**.

-cipher=*name* Specify the desired level of encryption. Valid values for *name* are HIGH, MEDIUM, LOW. If not specified, MEDIUM will be used. These values map to specific cipher suite names, which are determined by the version of OpenSSL being used. You can use the SLC option to the

GETFA routine to see which cipher suite is being used. See GETFA in the “System-Supplied Subroutines, Functions, and Classes” chapter of the *Synergy Language Reference Manual*. Used with **-cert** and **-encrypt**.

-encrypt*[=level]* Enable data packet encryption for *xf*Server or—if **-w** is specified—for *xf*ServerPlus. Valid values for *level* are MASTER and SLAVE. If *level* is not specified, slave encryption is used by default. Used with **-cert** and **-cipher**. See “Using Encryption” on page 3-8 for details on *xf*Server encryption or “Using Encryption” in the “Configuring and Running *xf*ServerPlus” chapter of the *Developing Distributed Synergy Applications* manual for details on *xf*ServerPlus encryption.

-h Display a list of all **rsynd** options.

-i*address* Bind the specified IP address to *xf*Server. Used when the machine has more than one IP address.

-l*filename* Specify the monitor log filename. Used with **-m**.

-m Start the monitor utility for *xf*Server. See “The Monitor Utility for UNIX” in the “General Utilities” chapter of *Synergy Language Tools* for information about running the monitor.

-n Run *xf*Server in non-secure mode. See also the **-u** option. Does not apply to *xf*ServerPlus.

-p*port* Specify the port on which *xf*Server or *xf*ServerPlus listens. The default port is 2330 for *xf*Server and 2356 for *xf*ServerPlus. *Port* overrides the SCSPORT environment variable.

-q Stop *xf*Server or *xf*ServerPlus without killing existing connections. If **-p** is not specified, the *xf*Server process on port 2330 will be stopped or—if **-w** is specified—the *xf*ServerPlus process on 2356 will be stopped. See also the **-c** option.

-rd*port[***:***timeout]* Enable remote debugging via Telnet for *xf*ServerPlus on the specified port. Used with **-w**. *xf*ServerPlus will listen on the specified port for the Telnet client; this port number must be different than the port that *xf*ServerPlus 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 *xf*NetLink–*xf*ServerPlus connection has been made. The default is 100 seconds. See the Usage section on page 3-45.

-s Run *xf*Server in secure mode. Does not apply to *xf*ServerPlus.

-t*time* Specify the time interval for monitoring. Used with **-m**.

-u*persona*

*xf*Server: Specify *persona* as the default user name or user name and password when running *xf*Server 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 “Modes” on page 3-26 for more information.

*xf*ServerPlus: Specify *persona* as the user name or user name and password under which to run children launched by *xf*ServerPlus. 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 current version of **rsynd**.

-w

Enable *xf*ServerPlus. See the *Developing Distributed Synergy Applications* manual for more information on *xf*ServerPlus.

Usage

You can place a space between an option and its argument if desired.

The following example shows how to start *xf*Server on port 2445 in secure mode:

```
rsynd -p 2445 -s
```

The following example shows how to start *xf*Server 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"
```

The following example shows how to start *xf*ServerPlus. Because we did not specify a port, *xf*ServerPlus 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 *xf*Server in non-secure mode with a default user account.

```
rsynd -w -u `setruser`
```

The following example shows how to start *xf*ServerPlus 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. *xf*ServerPlus 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 *Developing Distributed Synergy Applications* manual.

```
rsynd -rd 2445:60 -w -u ‘setruser‘
```

The following example shows how to start *xf*Server in secure mode on the default port and enable slave encryption. Because we did not specify **-cert** and **-cipher**, the default **.pem** filename will be used with a cipher level of medium.

```
rsynd -s -encrypt
```

The following example shows how to start *xf*Server in secure mode on the default port and enable master encryption. We set the cipher name to high and specify a **.pem** file that has a non-default name.

```
rsynd -s -encrypt=master -cipher=high -cert=CERT:myCertFile.pem
```

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 *xf*Server and enable slave encryption using the default **.pem** filename:

```
rsynd -s -cipher=low
```

The following example shows how to stop *xf*Server running on port 2445:

```
rsynd -p 2445 -q
```

VMS

/ALLOW_PRIVILEGED

Specify that user names that possess system-level privileges may be used as the default user account for *xf*Server or as the user name under which *xf*ServerPlus 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 and /ENCRYPT.

/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, which are determined by the version of OpenSSL being used.

You can use the SLC option to the GETFA routine to see which cipher suite is being used. See GETFA in the “System-Supplied Subroutines, Functions, and Classes” chapter of the *Synergy Language Reference Manual*. Used with /CERTIFICATE and /ENCRYPT.

/DEFAULT_USER=*user_name*

Specify the default user account for anonymous access to *xf*Server 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 “Modes” on page 3-35 for more information.)

/ENCRYPT=(MASTER|SLAVE|OFF)

Enable data packet encryption for *xf*Server or—if /XFPL_ENABLE is specified—for *xf*ServerPlus. Specifying OFF is the same as not specifying the /ENCRYPT option. Used with /CERTIFICATE and /CIPHER. See “Using Encryption” on page 3-8 for details on *xf*Server encryption or “Using Encryption” in the “Configuring and Running *xf*ServerPlus” chapter of the *Developing Distributed Synergy Applications* manual for details on *xf*ServerPlus encryption.

/FREE_POOL=*nnn*

Specify the number of *xf*Server 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 *xf*Server 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 logfile with /OUTPUT. If

OPERATOR is also specified, messages are sent to the console in addition to a logfile. See “Logging” on page 3-36 for additional information.

/NOSECURE

Specify that the server should run in non-secure mode. This is the default. Does not apply to *xf*ServerPlus. See also /SECURE.

/NOUSE_SYNRC

Specify that the *xf*Server file servers do 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 3-36 for additional information.

/PORT=*nnnn*

Specify the IP port on which *xf*Server or *xf*ServerPlus listens. The default port is 2330 for *xf*Server and 2356 for *xf*ServerPlus.

/SECURE

Specify that the server requires an encrypted password to be passed along with the user name of the client requesting service from *xf*Server. Does not apply to *xf*ServerPlus. The default is /NOSECURE.

/SHUTDOWN*[*=ALL*]*

Stop *xf*Server or *xf*ServerPlus without killing existing connections. For *xf*Server, if ALL is specified, existing connections are stopped as well. For more information, see “Stopping xfServer” on page 3-38.

/USE_SYNRC

Specify that the *xf*Server 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 *xf*ServerPlus and run *xf*ServerPlus server processes under the specified user name. See the *Developing Distributed Synergy Applications* manual for more information on *xf*ServerPlus.

/XFPL_FREE_POOL=*nnn*

Specify the number of *xf*ServerPlus 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 *xf*Server 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 *xf*Server process can have outstanding at any one time. The default is 800.

/EXTENT=*nnn*

Specify the maximum to which an *xf*Server 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 *xf*Server process can have open at any one time. The default is 256.

/MAXIMUM_WORKING_SET=*nnn*

Specify the maximum to which an *xf*Server 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 *xf*Server process can allocate. The default is 262136 on Itanium and 65536 on Alpha.

/PRIORITY=*nnn*

Specify the priority level at which an *xf*Server process runs. The default is 5.

The synxfpng Utility

The **synxfpng** utility can be used for troubleshooting network connections to *xf*Server. To verify that a client can access *xf*Server, run **synxfpng** from the client. **Synxfpng** uses *only* TCP/IP API calls and tests *only* whether the client can create and bind a socket to *xf*Server; it does not send packets to the server.

Synxfpng prints a report to the screen, showing each port tested. If there are errors, the TCP/IP error number is printed to the screen. This information can be used by your network administrator to resolve TCP/IP network socket communication problems. **Synxfpng** is installed in the dbl\bin directory.

The **synxfpng** utility can be used to test *any* TCP/IP connection, regardless of whether *xf*Server is involved. Use the verbose option to get detailed information that can be used for troubleshooting. See the **-x** option below if you want to test SQL OpenNet.

Syntax

`synxfpng` *[-option] [...] server_name*

Arguments

option

One or more of the following:

i*num* Ping the specified port/server *num* times.

p*port* Ping the specified port.

r*num* Ping *num* ports, starting with the port specified with the **-p** option.

v Verbose output (see sample below). Use this option to debug connection failures.

x Ping SQL OpenNet server connection.

server_name

The host name or IP address of the system to ping.

Discussion

By default, **synxfpng** pings 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.

If you do not specify a port with **-p**, **synxfpng** will use the SCSPORT, BEGPORT, and ENDPORT environment variables as follows:

- If the environment variable SCSPORT is set, that port is tested. If no service is found on that port, **synxfpng** will test up to 10 ports as described above.
- If the environment variable ENDPORT is set, **synxfpng** tries all ports from the default port 2330 (or 1958 when **-x** is used) through ENDPORT. (ENDPORT must be greater than the default port.)
- If the environment variables SCSPORT and ENDPORT are set, **synxfpng** tries all ports from SCSPORT through ENDPORT. (ENDPORT must be greater than SCSPORT.)
- If the environment variables BEGPORT and ENDPORT are set, **synxfpng** tries all ports from BEGPORT through ENDPORT. (ENDPORT must be greater than BEGPORT.) If SCSPORT is also set, it is ignored.

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, you've determined that there is a DNS lookup problem. Consult your network administrator for assistance in resolving the problem.

When used with the **-x** option, **synxfpng** pings port 1958, the default SQL OpenNet server port, by default. Use **synxfpng** with the **-x** and **-v** options to get the verbose error reporting that is unavailable with **vtxping**. **Synxfpng -x** uses the BEGPORT and ENDPORT environment variables as described in bullets two and four above; SCSPORT is not used.

Usage

- To ping the server named tiger on the default port (2330):

  ```
  synxfpng tiger
  ```

  Sample output:

  ```
  Pinging Synergy xfServer on host [tiger] port 2330
  Round trip (28 bytes) 0.64 ms
  ```

- To ping port 2440, located on the server named tiger, five times:

  ```
  synxfpng -p2440 -i5 tiger
  ```

- To ping ports 2355 through 2359 on the server named tiger:

  ```
  synxfpng -p2355 -r5 tiger
  ```

- To ping the server named tiger on the default 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 -- Complete 10.1.9.201

  Pinging Synergy/DE xfServer on host [tiger] port 2330

  Starting socket(AF_INET, SOCK_STREAM, IPPROTO_TCP) -- Completed
  Starting bind() -- Completed
  Starting connect() -- Completed
  Starting setsockopt(, SOL_SOCKET, SO_KEEPALIVE,,) -- Completed
  Starting setsockopt(, IPPROTO_TCP, TCP_NODELAY,,) -- Completed

  Round trip (28 bytes) 0.07 ms
  ```

The synckusr Utility

WIN

The **synckusr** utility can be used to diagnose *xf*Server problems that are related to user account setup.

Syntax

`synckusr` *[-option] [user_name] [service_name]*

Arguments

option

One of the following:

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 *xf*Server service for which you want to look up the default file path. Used with **-s**.

Discussion

If there is an *xf*Server 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 *xf*Server. Then, run **synckusr** on the server machine with the **-p** option. This will tell you which user name and password *xf*Server 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 **synckuser** on the server with the **-s** and **-p** options. This will display the default file directory for the specified *xf*Server 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 and System Options*.

Usage

- To display the user name and password that will be passed through the client connection to *xf*Server, 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 and the Windows security ID (SID).

  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>
User account is found in domain: ABC_CORP
User SID is: S-1-5-21-516626026-640904803-9522986-1071
```

  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
User account is found in domain: ABC_CORP
User SID is: S-1-5-21-516626026-640904803-9522986-1071
```

  Sample output when RUSER is not set and the user name is found on the local machine:

```
RUSER not set. <mark> from GetUserName system call (implied RUSER)
No Password
User account is found in local computer: TIGER
User SID is: S-1-5-21-487314201-320540311-1472111-1003
```

- To perform a user account lookup and authenticate it, run **synckusr** on the *xf*Server 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. 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 *xf*Server in secure mode.

  For example, on the *xf*Server 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 perform a user account lookup and display where the user name was found, run **synckusr** on the *xf*Server 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 *xf*Server 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 *xf*Server 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, run **synckusr** with the **-p** and **-s** options on the *xf*Server 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 *xf*Server machine you'd enter

  ```
  synckusr -p Mark -s SynSrv
  ```

  Sample output when RSFILPATH is set for a specific instance of *xf*Server:

  ```
  Password: *********
  Default directory: <c:\Synergy\Data> from RSFILPATH in server Synrc
  (HKLM)
  User account is found in local computer: ABC_CORP
  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
```

See also

- RUSER in the “Environment Variables” chapter of *Environment Variables and System Options*.
- RSFILPATH in the “Environment Variables” chapter of *Environment Variables and System Options*.
- “The setruser Utility” on page 3-57.

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.
- On UNIX, it returns the encoded string, which you can then use to set RUSER (see the Usage section on page 3-59).

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**. 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.

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** and uses the setting found there. If RUSER is not set in **HKEY_CURRENT_USER**, then the runtime checks **HKEY_LOCAL_MACHINE**.

When RUSER is set in the **Synrc** node in the registry, as is required when configuring *xf*ServerPlus 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 *xf*Server 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.

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.

The **-n** option prompts you for the user name and password, and then displays the encoded string to the screen without updating the registry.

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, *xf*Server will find the account on the local machine first, and then display an error because the password doesn’t match.

Usage

On UNIX, run **setruser** and enter the user name and password when prompted. 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 and System Options*.

Troubleshooting *xf*Server

Here are some suggestions in case you run into problems configuring *xf*Server.

General troubleshooting tips

When you are experiencing problems with *xf*Server, 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 Synergy Configuration Program to turn on additional logging. (See page 3-14.) 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 Language Tools*.)
- **UNIX:** By default, **rsynd** logs start/stop information and critical errors. (See page 3-28.) The Monitor utility may also be of some assistance when troubleshooting. (See “The Monitor Utility for UNIX” in the “General Utilities” chapter of *Synergy Language Tools*.)
- **OpenVMS:** By default, **rsynd** logs errors and some status messages. You can use the /LOG_LEVEL=FULL qualifier for additional logging. (See page 3-36.)

Start-up errors (Windows)

When I start the *xf*Server 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 3-50 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 *xf*Server.
- 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 *xf*Server 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 *xf*Server. See “The synckusr Utility” on page 3-53 for details. Refer to the following sections for information on how and where RUSER is set:

- “The setruser Utility” on page 3-57
- RUSER in the “Environment Variables” chapter of *Environment Variables and System Options*
- For Windows, “Understanding xfServer security” on page 3-16
- For UNIX, “Modes” on page 3-26
- For OpenVMS, “Modes” on page 3-35

File permissions (UNIX)

How do I get *xf*Server 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.

On SCO, you must *always* set UMASK; otherwise, files will be created with permissions for read and write by owner only.

File access (Windows)

What should I do when I get a “file not found” message when starting an *xf*Server service?

Remove the *xf*Server 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 3-23 and “Starting xfServer” on page 3-22.

I’ve created a local user account on my server. Why can’t I view the profile?

Until the user has physically logged on to the system, you cannot view that user’s profile. Once the user has logged on, you can view the profile information from two files. Normally, these two files are located in the %WINDIR%\system32\config directory. You can also view the user profile information in the registry. The file path is

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\
ProfileList\user’s_SID
```

Why do I get a protection error accessing files on *xf*Server, 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 *xf*Server, run the **synckusr** utility on your *xf*Server machine. See “The synckusr Utility” on page 3-53 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.

***xf*Server 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 page 3-50 for more information on **synxfpng**.

Rsynd: shutting down and timing out

Connections to the *xf*Server daemon, **rsynd**, are normally closed down by the client through normal exit processing, thereby freeing up 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 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.

Unfortunately, the wait time for the KEEPALIVE timer is normally about two hours on most systems. We recommend you set it to about 10 minutes.

- Windows: The KEEPALIVE timer is set with the KeepAliveTime registry entry. You can run the **syncksys** utility with the **-a** option to view your current KeepAliveTime setting. See the operating system documentation for your version of Windows for details on setting this registry entry.
- UNIX: Some UNIX systems allow changes to the KEEPALIVE timer. See your operating system documentation or supplier for details.
- OpenVMS: The amount of wait time is much less on OpenVMS systems and can be set in TCPIP. See your 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, *xf*ODBC, and SQL OpenNet.

SQL Connection is an API that enables you to access third party databases, such as Oracle, from a Synergy application. *xf*ODBC 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 *xf*ODBC 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 *xf*ODBC Client, which is a thinner client that does not require the installation of Core Components. (For details on the *xf*ODBC 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 *xf*ODBC, as well as how to configure SQL Connection for stand-alone or client use and *xf*ODBC for client use (Windows only). For information about using SQL Connection and *xf*ODBC, see the product manuals: *SQL Connection Reference Manual* and *xfODBC User's Guide*.

SQL Connection and *xf*ODBC on Windows

This section explains how to configure SQL Connection and *xf*ODBC 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 4-4 for server set-up and "SQL Connection: configuring client or stand-alone access" on page 4-8 for client set-up.
- To run **SQL Connection stand-alone**, see "SQL Connection: configuring client or stand-alone access" on page 4-8. (Stand-alone is defined as any connection that doesn't use SQL OpenNet.)
- To run ***xf*ODBC in a client/server configuration**, see "SQL Connection and xfODBC: using SQL OpenNet" on page 4-4 for server set-up and "xfODBC: testing the network connection for client access" on page 4-10 for client set-up.
- To run ***xf*ODBC stand-alone**, see "xfODBC: stand-alone access" on page 4-10.

Understanding SQL OpenNet on Windows

On Windows, you can run SQL OpenNet with either the **vtxnetd** program (the default) or the **vtxnet2** program. Command-line options and environment variables are identical for the two programs, and 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 4-25.

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., **vtx0**) 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_odbc** 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 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., **!vtx0**) 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 4-7.

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 4-11 and "Customizing the opennet.srv file" on page 4-7 for more information.

SQL Connection and *xf*ODBC: using SQL OpenNet

To use SQL OpenNet with SQL Connection or *xf*ODBC, you need to do the following on your SQL OpenNet server machine:

- Select the drivers to use for Oracle and SQL Server (SQL Connection only).
- Specify the SQL OpenNet port number in the TCP/IP services file.
- Start and test the server.

Selecting drivers for Oracle and SQL Server

Because SQL Connection offers several database drivers for Oracle and SQL Server, a default driver is selected for each of these databases during installation. If you use an Oracle or SQL Server database, use the Synergy Configuration Program to verify that the selected driver is the one you want to use, and change it if necessary. This driver is used when a connect string includes the **VTX0** or **VTX12** driver names for SQL OpenNet connections or the "oracle" or "sqlserver" keywords for direct connections. (For more information on choosing a driver, see "Building Connect Strings" in the "Creating SQL Connection Programs" chapter of the *SQL Connection Reference Manual*.)

1. Start the Synergy Configuration Program (from the Start menu select Programs > Synergy/DE > Utilities > Synergy Configuration Program).
2. Go to the Connectivity Series tab.
3. Select the driver that you want to use.
4. Click Apply.

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 c:\winnt\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 4-7 and "The vtxnetd and vtxnet2 Programs" on page 4-25.
- Specify the port on the client. For SQL Connection, see page 4-8 (Windows), page 4-15 (UNIX), or page 4-22 (OpenVMS). For *xf*ODBC, 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 in the administrator group.

1. Before starting SQL OpenNet, you must first register the **SynSQL** service. Do one of the following:

   - Start the Synergy Configuration Program, go to the Connectivity Series tab, and click the Add Service button. (Note: 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 4-29 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 4-11.

   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 and stand-alone)" on page 4-9 for information on the example programs. See chapter 2 for licensing information.

Stopping and removing SQL OpenNet

When you stop SQL OpenNet, no new connection can be made, but existing connections are not stopped. Only the third-party applications that are connected to the Synergy data can stop existing processes. You must stop the service before you can remove (unregister) it.

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, 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.

Information on using **vtxnet2** instead of the default **vtxnetd**, defining environment variables, caching, and changing the polling interval is below. For information on the following, see the **vtxnetd** and **vtxnet2** program syntax on page 4-25:

- Using a port other than the default
- 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 page 4-3 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 *xf*ODBC 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 or stand-alone and how to test your configuration.

Configuring SQL Connection as a client

1. Add the following line to your TCP/IP **services** file in c:\winnt\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 4-29 for more information.

Configuring SQL Connection as stand-alone

Because SQL Connection offers several database drivers for Oracle and SQL Server, a default driver is selected for each of these databases during installation. If you use an Oracle or SQL Server database, use the Synergy Configuration Program to verify that the selected driver is the one you want to use, and change it if necessary. (If you're not using an Oracle or SQL Server database, skip to "Testing SQL Connection (client and stand-alone)" on page 4-9.) The default driver is used when a connect string includes the **VTX0** or **VTX12** driver names for SQL OpenNet connections or the "oracle" or "sqlserver" keywords for direct connections. (For more information on choosing a driver, see "Building Connect Strings" in the "Creating SQL Connection Programs" chapter of the *SQL Connection Reference Manual*.)

1. Start the Synergy Configuration Program (from the Start menu select Programs > Synergy/DE > Utilities > Synergy Configuration Program).
2. Go to the Connectivity Series tab.
3. Select the driver that you want to use.
4. Click Apply.

Testing SQL Connection (client and 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_odbc**).

      The syntax for *connect_string* depends on the driver. For a list of driver names and the connect syntax to use with them, see "Connect string syntax" 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_ODBC
      ```

    - On a stand-alone configuration, use this connect string:

      driver_keyword:connect_string

      where *driver_keyword* is the keyword (e.g., oracle) for 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 "Connect string syntax" in the "Creating SQL Connection Programs" chapter of the *SQL Connection Reference Manual* for complete information.

      For example, to connect to an Oracle database, the connect string consists of the user ID and password. So, you might enter

      ```
      set SQL_CONNECT = oracle: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.

*xf*ODBC: testing the network connection for client access

After configuring SQL OpenNet on the server and installing Connectivity Series or *xf*ODBC Client on the client, you should test the network connection before attempting to access data using *xf*ODBC. 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 (1958) 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 4-29 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 *xf*ODBC. See "Using the Sample Database as a Tutorial" in the *xfODBC User's Guide*.

*xf*ODBC: 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 *xf*ODBC. 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 4-7). **Sqld** also polls **vtxnetd/vtxnet2** periodically to verify that the program is still running (see “Changing the polling interval” on page 4-7). 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` *[-option] [...]*

Arguments

option

One or more of the following:

h	Display a list of options.
l	Log debug information about **SynSQL** startup in the Windows event log. Used with **-r** or **-rs**.
q	Stop the service **SynSQL**.
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.

When you stop the **SynSQL** service, no new connections can be made, but existing connections are not stopped. Only the third-party applications that are connected to the Synergy data can stop existing processes.

SQL Connection and *xf*ODBC on UNIX

This section explains how to configure SQL Connection and *xf*ODBC 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 4-13 for server set-up and "SQL Connection: configuring client or stand-alone access" on page 4-15 for client set-up.
- To run **SQL Connection stand-alone**, see "SQL Connection: configuring client or stand-alone access" on page 4-15. (Stand-alone is defined as any connection that doesn't use SQL OpenNet.)
- To run ***xf*ODBC in a client/server configuration**, see "xfODBC: using SQL OpenNet" on page 4-18 for UNIX server set-up and "xfODBC: testing the network connection for client access" on page 4-10 for Windows client set-up.

To use Connectivity Series products, you must run the **setsde** script. This script enables the Synergy/DE environment for the products you have installed. If you have not yet run **setsde**, do so now; see "Environment setup" on page 1-6 for additional information.

The **setsde** script can be run on 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 4-25 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 run the Oracle script **oraenv**.

1. Install Connectivity Series from your distribution and run the **setsde** script (see the note on page 4-12).
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 4-14.
   - 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 4-16.

   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 4-25 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 for licensing information.

Starting and stopping SQL OpenNet for SQL Connection

Starting SQL OpenNet

Follow these instructions to configure and run 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 run the Oracle script **oraenv** before starting SQL OpenNet—even if you've already run 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. Run 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 the **setsde** and **startnet** scripts run *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 4-25 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 4-29 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 run the **setsde** script (see the note on page 4-12).
- 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 run 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 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.

*xf*ODBC: using SQL OpenNet

To use SQL OpenNet for *xf*ODBC access, you need to

- configure SQL OpenNet for *xf*ODBC.
- start SQL OpenNet.

Configuring SQL OpenNet for *xf*ODBC

Before configuring SQL OpenNet, you must have TCP/IP installed, configured, and working properly.

1. Install Connectivity Series from your distribution and run the **setsde** script (see the note on page 4-12).
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 4-25 for details on the **vtxnetd** command-line options.

Starting and stopping SQL OpenNet for *xf*ODBC

Follow these instructions to configure and run the **setodbc** script, which kills and restarts SQL OpenNet, and can also be used to set *xf*ODBC environment variables. The **setsde** script can be run on any UNIX shell *except* **csh**.

1. Move to the synodbc directory. For example:

   ```
   cd /usr/synergyde/connect/synodbc
   ```

2. (optional) Edit the set-up script file, **setodbc**, to include *xf*ODBC environment variables. See "Setting environment variables" 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. Run 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 *xf*ODBC. However, if you do use it, and you see this error when running the **setodbc** script

   ```
   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** was subsequently run, there was a case mismatch, resulting in an error.

   To correct this situation, set DBLCASE to blank and run **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 run *after* License Manager is started.

Your *xf*ODBC server component is now ready to use. For information about using *xf*ODBC, 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 4-25 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 4-29 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 *xf*ODBC on OpenVMS

This section explains how to configure SQL Connection and *xf*ODBC 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 4-20 for server set-up and "SQL Connection: configuring and testing client or stand-alone access" on page 4-22 for client set-up.
- To run **SQL Connection stand-alone**, see "SQL Connection: configuring and testing client or stand-alone access" on page 4-22. (Stand-alone is defined as any connection that doesn't use SQL OpenNet.)
- To run ***xf*ODBC in a client/server configuration**, see "xfODBC: using SQL OpenNet" on page 4-23 for OpenVMS server set-up and "xfODBC: testing the network connection for client access" on page 4-10 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 web site) 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 4-25 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 4-20).

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: ORACLE81, ORACLE9, ORACLE10, SYBASE-DB, RDB6, 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 4-22). For details on the **vtxnetd** command-line options, see "The vtxnetd and vtxnet2 Programs" on page 4-25.

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, the default port (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 4-29 for more information. For SQL Connection, you must license your database drivers before you can run the example programs. See chapter 2 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 (1958) 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 4-21.
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.

*xf*ODBC: using SQL OpenNet

To use SQL OpenNet for *xf*ODBC access, you need to

- configure SQL OpenNet for *xf*ODBC (optional).
- start SQL OpenNet.

Configuring SQL OpenNet for *xf*ODBC

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 4-24.

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 4-25.

Starting and stopping SQL OpenNet for *xf*ODBC

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, the default port (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 4-29 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, the default port (1958) is used.

The vtxnetd and vtxnet2 Programs

Vtxnetd and **vtxnet2** are service programs that implement SQL OpenNet. **Vtxnet2** is available only on Windows. The syntax and options for the two programs are the same. 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 4-7.
- On UNIX, the command line to start **vtxnetd** is located in the **startnet** script. It can also be run from the command line. See "Starting and stopping SQL OpenNet for xfODBC" on page 4-18.
- On OpenVMS, the line for starting the database driver is located in the input file **NET.COM** (see page 4-24).

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. See "Starting the service manager with a password (-a[L])" on page 4-26.

-f*filename* Use the specified file to check inbound service requests. Not available on OpenVMS. See "Rejecting invalid service requests (-f)" on page 4-26.

-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. See "Using a log file (log, log2)" on page 4-27.

log2 Output a connection log that does not include **vtxping** events. See "Using a log file (log, log2)" on page 4-27.

-p*nnnn* Listen for requests on port *nnnn*. See "Specifying a port number (-p)" on page 4-27.

-s*n* Specify Windows thread stack size for **vtxnetd** in kilobytes. See "Specifying the amount of memory used by vtxnetd on Windows (-s)" on page 4-28.

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.
- 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.

Rejecting invalid service requests (-f)

The **-f** option is not available on OpenVMS.

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

```
/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. For example, if the process ID is 175, and the service manager is started in c:\synergyde\connect, the log file created is **c:\synergyde\connect\tcm_175.log**.

Other types of logging are available. For information on SQL Connection logging, see “Troubleshooting and Error Logging” in the “Error Logging and Messages” chapter in the *SQL Connection Reference Manual*. For information on *xf*ODBC logging, see “Error Logging” in the “Data Access Errors and Error Logging” chapter of the *xfODBC User’s Guide*.

Specifying a port number (-p)

The default port used by the service manager is defined in the **services** file (on Windows and UNIX) and is hard-coded to 1958 on OpenVMS. (For details on setting the port number in the **services** file, refer to the operating system–specific sections of this chapter.) The default port can be overridden with the **-p** option.

Vtxping and **vtxkill** also use the port defined in the **services** file (Windows and UNIX) or hard-coded (OpenVMS), unless you pass a port number when you run them. They do not use the port specified on the **vtxnetd/vtxnet2** command line.

If you use a non-default port for **vtxnetd/vtxnet2**, you must also change the port number on the client. For SQL Connection, see page 4-8 (Windows), page 4-15 (UNIX), or page 4-22 (OpenVMS). For *xf*ODBC, 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***n* 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
```

Note, however, that 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.

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 3-50.

Syntax

vtxping *[-option] server_name [server_name…]*

Arguments

option

One or more of the following:

m*num* Retry a maximum of *num* times. The default is 3.

p*port* 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

When you run **vtxping** without any options, it displays usage information.

By default, **vtxping** uses the port specified in the services file on Windows and UNIX and port 1958 (hard-coded) on OpenVMS.

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

Overview

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 insert the installation CD and install the components that your application requires. 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
- *xf*NetLink Java, *xf*NetLink COM, and *xf*NetLink .NET
- *xf*ODBC Client
- Data Provider for .NET
- Synergy/DE Client
- Online Manuals

For information on installing Synergy/DE Client, refer to the installation instructions that accompanied your distribution.

This chapter only discusses the command line options for installing Synergy products.

For detailed information on deploying *xf*NetLink applications, refer to the section of the *Developing Distributed Synergy Applications* manual that applies to your client.

For detailed information on deploying *xf*ODBC applications, see chapter 4 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, installation location, 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 must pass properties to specify the required information. See "Properties" on page 5-6.
- For installation location and component selection, the current values are used on an upgrade. On a new installations, default values are used. See "Default installation location and components", below.

Regardless of the type of installation, the installed Synergy product is included in the Add/Remove Programs list (Programs and Features on Vista/2008 and higher) 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 reduced, basic, 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.

Default installation location and components

The table below shows the default installation locations for the various Synergy products. The defaults are used only on a new installation; an upgrade will be installed in the current location.

Product	Default installation directory
Synergy/DE	<Program Files>\Synergex\SynergyDE
*xf*NetLink Java	<Program Files>\Synergex\xfNLJava
*xf*NetLink COM	<Program Files>\Synergex\xfNLCOM
*xf*NetLink .NET	<Program Files>\Synergex\xfNLNET
*xf*ODBC Client	<Program Files>\Synergex\SynergyDE\Connect
Data Provider for .NET	<Program Files>\Synergex\SynergyDE Data Provider[a]
Online Manuals	<Program Files>\Synergex\SynergyDE\Manuals

[a] This directory cannot be changed even with a full UI installation.

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
- *xf*Series
- Files for shared installation (Windows Server 2003, 2008, and 2008 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.

On Vista and higher, 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. However, if you perform a basic or reduced UI installation from a non-elevated command prompt, the installation completes, and the products run properly, but the uninstallation will fail.

All operating system requirements (such as service packs) 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., **101sde931.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 from the CD or by unpacking the web download: **setup.exe**, **setup.ini**, ***filename*.msi** (e.g., **101sde931.msi**), **splashver.bmp**, **0x0409.ini**.

The splash screen always displays when using **setup**, regardless of the user interface level specified.

Using the Windows Installer directly

To install using the Windows Installer directly, run **msiexec** and specify the **.msi** filename (e.g., **101sde931.msi**) on the command line, along with the desired **/q** options and licensing properties (if required).

You need only the **.msi** 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

For basic, reduced, and no UI installations, you must pass properties to specify the license information on a new installation. For a full UI installation, you can use the properties to pass information that you want to display by default in the License Configuration dialog box, but users can override the defaults. License properties are required only for a Synergy/DE installation.

Properties are case sensitive.

LICENSETYPE. Specify either Client or NonClient (case sensitive).

SERVERNAME. The name of the license server machine if LICENSETYPE is Client.

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 the examples below.

If you do not pass license properties (or specify incomplete information) to a reduced, basic, 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.

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

A. This example shows a silent installation of Synergy/DE using **msiexec**, with client licensing.

```
msiexec /i 101sde931.msi /qn LICENSETYPE=Client SERVERNAME=Tiger
```

B. This example shows a silent installation of *xf*NetLink .NET using **msiexec**. (License properties are not required for an *xf*NetLink installation.)

```
msiexec /i 101nn931.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.

```
setup /v"/qb+! LICENSETYPE=Client SERVERNAME=Tiger"
```

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 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 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 101sde931.msi /qn /lv* %temp%\uninstall.log
```

The **/lv*** option produces a logfile. You may want to include this option when doing a silent install (or uninstall) because if an error is encountered, the installation may abort without displaying the error to the screen. The logfile option is needed only on Windows XP and Server 2003 platforms; on all other Windows platforms a logfile is created automatically by Windows Installer Service 4 or higher and placed in the user's %TEMP% directory.

G. This example shows a silent uninstall 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 101sde931.msi /qn /lv* %temp%\uninstall.log"
```

Index