

# Tru64 UNIX Best Practice

---

## Using Solaris Compatibility Libraries When Migrating Solaris Applications to Tru64 UNIX

June 2001

**Product Version:** *Solaris Compatibility Libraries Version 1.1*

**Operating System and Version:** *Tru64 UNIX versions 4.0D and later*

This Best Practice describes how to quickly begin using the Solaris Compatibility Libraries (SCL) when migrating Solaris applications to the Compaq Tru64™ UNIX operating system.

---

**Compaq Computer Corporation**  
**Houston, Texas**



---

# Contents

## ***Using Solaris Compatibility Libraries When Migrating Solaris Applications to Tru64 UNIX***

Is This Best Practice Right for You? .....	1
Before You Begin .....	2
Downloading the Software .....	3
Applying the Best Practice .....	3
Installing SCL .....	3
Building Threads Applications .....	5
Using the RPC Library .....	5
Compile Using SCL RPC .....	6
Using the Miscellaneous Components .....	8
Verifying Success .....	8
Troubleshooting .....	8
Alternative Practices .....	9
Comments and Questions .....	10
Legal Notice .....	10



---

# Using Solaris Compatibility Libraries When Migrating Solaris Applications to Tru64 UNIX

This Best Practice describes how to quickly begin using the Solaris Compatibility Libraries (SCL) when migrating Solaris applications to the Compaq Tru64™ UNIX operating system. SCL allows you to more easily port applications that use non-standard, proprietary Solaris application programming interfaces (APIs) to Tru64 UNIX. SCL enables application source to be ported with little or no change and prevents the need for writing your own implementation of the Solaris functionality.

See the Tru64 UNIX Best Practices Web page for more information about Best Practices documentation.

## Is This Best Practice Right for You?

Not all Best Practices apply to all configurations, so you must be sure that it is appropriate for your system and circumstances. To use this Best Practice, you must meet the requirements described in the following table:

Requirement	Description
Operating System	Tru64 UNIX Version 4.0D and higher, including systems with operating system patches applied.
Impact on Availability	None.
Additional Requirements	Required subsets for software development include: OSFSDE $nnn$ , OSFPGMR $nnn$ , OSFINCLUDE $nnn$ , OSFLIB $nnn$ , and OSFCMPLRS $nnn$ , where $nnn$ represents the version number, which varies according to which release of Tru64 UNIX you have installed. You must have root login privileges.

If you do not meet the previous requirements, see *Alternative Practices* for information.

## Before You Begin

Before you apply the Best Practice for using Solaris Compatibility Libraries, you must understand some background information and perform some preliminary tasks.

- You are installing Version 1.1 of the Solaris Compatibility Libraries.
- Everything will be installed in the default directories, in `/usr/opt/SCL110`. A link is created from `/usr/opt/solcomplib` to `/usr/opt/SCL110`. The `/usr/opt/solcomplib` directory is known as the SCL root directory, and application make files and build commands should use this with the `-L` and `-I` compiler options.
- For the installation, you are running as root.
- After installation, you can find detailed documentation, including reference pages and a user's guide, in the `/usr/opt/solcomplib/doc` directory.
- SCL functions fall into three categories: Solaris Threads Library, Remote Procedure Calls, and miscellaneous components.
  - The Solaris Threads Library (STL) provides an implementation of most of the Solaris threads APIs for Tru64 UNIX. It is essentially a layer of software that sits between the application program and the operating system's underlying POSIX threads implementation. With STL, most Solaris thread functions are implemented by calling the equivalent POSIX threads function.
  - The Remote Procedure Calls (RPC) component of the Solaris Compatibility Libraries is based on the SunSoft Transport-Independent RPC (TI-RPC) v2.3. The Tru64 UNIX operating system ships with a sockets-based version of Open Network Computing Remote Procedure Calls (ONC RPC) that is derived from an earlier distribution of code from Sun Microsystems, Inc. The SCL version of RPC provides enhanced `rpcgen` and `rpcinfo` utilities, as well as a version of `rpcbind` to act as a replacement for the standard portmapper. The two utilities and the portmapper are enhanced versions of those that ship with Tru64 UNIX as part of ONC RPC.
  - The miscellaneous library component of SCL comprises various groups of Solaris functions that are not standard on Tru64 UNIX, such as signal name, high-resolution timers, and asynchronous

I/O routines. Implementation is provided within the SCL-supplied shared and static libraries, or solely within the SCL-supplied header files using macros.

For detailed information about SCL functions, see the *Solaris Compatibility Libraries for Compaq Tru64 UNIX User's Guide*.

Porting considerations, such as 32-bit to 64-bit issues, build command differences, and data ordering (big-endian vs. little-endian) are not covered by SCL. For more information about these issues, see the *Sun Solaris to Compaq Tru64 UNIX Porting Guide*.

## Downloading the Software

You can download SCL from the following location:

<http://www.tru64unix.compaq.com/complibs/download.htm>

You need 14 MB of free disk space for the installation.

SCL Version 1.1 includes the following features:

- Solaris Threads Library component
- Remote Procedure Calls component
- Miscellaneous library components
- Complete SCL source code and development environment
- Complete function and command reference documentation in man and HTML formats
- Installation using the Tru64 UNIX `setld` subset management utility

## Applying the Best Practice

Before you begin using the Solaris Compatibility Libraries to migrate Solaris applications to Tru64 UNIX, be sure to follow the recommendations in *Before You Begin*.

## Installing SCL

Follow these steps to install the SCL:

1. Make a directory for the SCL kit:  

```
# mkdir directory/sclkit
```

2. Move the SCL kit that you downloaded to the directory for the SCL kit:

```
# mv SCL110kit.tar.gz directory/sclkit
```

3. Change the directory to the SCL kit directory:

```
# cd directory/sclkit
```

4. Unzip the compressed file:

```
# gunzip SCL110kit.tar.gz
```

5. Extract the SCL tar file:

```
# tar xvf SCL110kit.tar
```

6. Install the software:

```
# setld -l . SCLBASE110 SCLDOC110 SCLSOURCE110
```

Take the defaults (press ENTER) for all questions and prompts.

7. Set the basic environment variables:

- NLSPATH, the path to the SCL message catalog
- LD\_LIBRARY\_PATH, the path to the SCL shared library files
- PATH, the path to the binary files for the SCL commands and utilities
- MANPATH, the path to the SCL reference pages

The environment variables must be set correctly, so that the loader can find the SCL libraries and message-catalog file.

If you use the Bourne, Korn, or POSIX shell, enter the following commands:

```
# export NLSPATH=$NLSPATH:/usr/opt/solcomplib/lib/nls/msg/%N
# export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/opt/solcomplib/shlib
# export PATH=$PATH:/usr/opt/solcomplib/bin
# export MANPATH=$MANPATH:/usr/opt/solcomplib/man
```

Modify your `.profile` file to set these variables at login.

If you use the C or tcsh shell, enter the following commands:

```
# setenv NLSPATH /usr/opt/solcomplib/lib/nls/msg/%N
# setenv LD_LIBRARY_PATH /usr/opt/solcomplib/shlib
# setenv PATH /usr/opt/solcomplib/bin:$PATH
# setenv MANPATH /usr/opt/solcomplib/man:$MANPATH
```

Modify your `.cshrc` or `.login` file to set these variables at login.

## Building Threads Applications

To build a Solaris thread application on Tru64 UNIX, you need to define the location of the Solaris thread files. The Solaris Threads Library component of the SCL includes two header files, `synch.h` and `thread.h`, and the shared object library `libthread.so`.

The default location for `libthread.so` is `/usr/opt/solcomplib/shlib`. The default location for the header files `thread.h` and `synch.h` is `/usr/opt/solcomplib/include`.

1. To compile a program named `prog.c` using the SCL Threads Library, enter the following command:

```
# cc -c -pthread -I/usr/opt/solcomplib/include prog.c
```

You use the `-pthread` option in the `cc` command line to inform the compiler that your program is multi-threaded. Doing this is equivalent to using the `-mt` compiler option on Solaris. The compiler option `-I/usr/opt/solcomplib/include` informs the compiler where to find the thread header files.

2. To link the file, enter the following command:

```
# cc prog.o -L/usr/opt/solcomplib/shlib -lthread -o prog
```

You use the `-L` link editor option to change the library directory search order for shared object libraries, so that `/usr/opt/solcomplib/shlib` is searched. You use the `-l` option to specify that the SCL thread library is to be used.

For an example of a makefile, see `/usr/opt/solcomplib/examples/threads/makefile`.

## Using the RPC Library

To configure your system for SCL RPC, you must perform the following tasks:

1. Create the `scl_netconfig` file.

On a Solaris system, the file `/etc/netconfig` provides information about XTI name and address mapping. To provide this information on Tru64 UNIX, enter the following command:

```
# cp /usr/opt/solcomplib/examples/rpc/scl_netconfig  
/usr/opt/solcomplib/etc/scl_netconfig
```

2. Modify the `/etc/services` file to define `scl_rpcbind` Internet service. You do this by adding lines to the `/etc/services` file, such as the following:

```
scl_rpcbind      1112/udp      # Use port 1112 for SCL RPC library
scl_rpcbind      1112/tcp      #
```

3. Run `rpcbind`, the server that converts RPC program numbers into universal addresses. Enter the following command:

```
# /usr/opt/solcomplib/bin/rpcbind
```

---

**Note**

---

If you reboot, you will need to rerun `rpcbind`.

---

4. Test the SCL RPC configuration by entering the following command:

```
# /usr/opt/solcomplib/examples/rpc/test_sclrpc_config
```

The results should be similar to the following:

```
Testing for service name 'scl_rpcbind'....OK
```

```
Testing for existence of scl_netconfig file....OK
```

```
Looking for SCL RPC shared objects.. 0.00166|0001: SCL Logfile initialized..OK
```

```
Testing APIs for scl_netconfig file..
```

```
..found TCP entry..
```

```
..found UDP entry..
```

```
..Found 2 valid entries in scl_netconfig file
```

```
Testing access to XTI devices..
```

```
..succeeded in opening /dev/streams/xtiso/tcp+
```

```
..succeeded in opening /dev/streams/xtiso/udp+
```

```
.. 2 XTI devices OK
```

```
Total errors found : 0
```

```
0.90966|0001: SCL Logfile being closed
```

If errors are found, make sure that

`/usr/opt/solcomplib/etc/scl_netconfig` exists

and that entries for `scl_rpcbind` appear in `/etc/services`.

## Compile Using SCL RPC

The examples in this section assume the following file names:

`prog.x`                      RPC specification file

server.c                   The server application source file

client.c                   The client application source file

To compile using SCL RPC, follow these steps:

1. Use the rpcgen compiler to process the application using the application's RPC specification files. For example:

```
# /usr/opt/solcomplib/bin/rpcgen prog.x
```

The rpcgen compiler generates the following files:

- prog.h
- prog\_svc.c
- prog\_clnt.c
- prog\_xdr.c

2. Compile the client/server application components:

```
# cc -DXOPEN_SOURCE -pthread -D_SCL_ -std0  
-I/usr/opt/solcomplib/include -c client.c
```

```
# cc -DXOPEN_SOURCE -pthread -D_SCL_ -std0  
-I/usr/opt/solcomplib/include -c server.c
```

```
# cc -DXOPEN_SOURCE -pthread -D_SCL_ -std0  
-I/usr/opt/solcomplib/include -c prog_clnt.c
```

```
# cc -DXOPEN_SOURCE -pthread -D_SCL_ -std0  
-I/usr/opt/solcomplib/include -c prog_svc.c
```

```
# cc -DXOPEN_SOURCE -pthread -D_SCL_ -std0  
-I/usr/opt/solcomplib/include -c prog_xdr.c
```

3. Link the client/server application components:

```
# cc -DXOPEN_SOURCE -pthread -D_SCL_ -std0 client.o  
prog_clnt.o prog_xdr.o /usr/opt/solcomplib/shlib/librpc.so  
/usr/opt/solcomplib/shlib/libthread.so  
/usr/opt/solcomplib/shlib/libsolutil.so -lxti -o prog_client
```

```
# cc -DXOPEN_SOURCE -pthread -D_SCL_ -std0 server.o  
prog_svc.o prog_xdr.o /usr/opt/solcomplib/shlib/librpc.so  
/usr/opt/solcomplib/shlib/libthread.so  
/usr/opt/solcomplib/shlib/libsolutil.so -lxti -o prog_server
```

For an example of a Makefile, see  
`/usr/opt/solcomplib/examples/rpc/square/Makefile`.

4. Execute the server application.
5. Execute the client application.

## Using the Miscellaneous Components

For a list of these components and how to use them, see the *Solaris Compatibility Libraries for Compaq Tru64 UNIX User's Guide*.

## Verifying Success

After you apply the Best Practice for using Solaris Compatibility Libraries to migrate Solaris applications to Tru64 UNIX, you can verify whether it was successful by trying the following tests:

- Execute an example.

You can test the success of a build by executing the examples or the test programs that are located in `/usr/opt/solcomplib/examples/threads` and `/usr/opt/solcomplib/examples/rpc`. Copy the test files to a test directory. Change your directory to the test directory and run the make command.

- Access the reference pages.

Reference page help is available by entering the command `man`, followed by the filename, command name, or function name. For example, enter:

```
# man thr_create
```

Reference page output will provide information about the `thr_create` routine.

If you determine that the Best Practice was not successful, see *Troubleshooting* for information about identifying and solving problems.

## Troubleshooting

If you determine that the Best Practice was not successful, as described in *Verifying Success*, use the following table to identify and solve problems:

Problem	Possible Solutions
Cannot access reference pages.	<p>SCL provides reference pages for all implemented functions and commands. These reference pages reside in the man directory hierarchy beneath the SCL root directory. For the man command to search the SCL man directory, you need to define or modify the MANPATH environment variable. In the C or tcsh shell, you can do this as follows:</p> <pre># setenv MANPATH /usr/opt/solcomplib/man:\$MANPATH</pre> <p>In the Bourne, Korn, or POSIX shell, you can do this as follows:</p> <pre># export MANPATH=\$MANPATH:/usr/opt/solcomplib/man</pre> <p>The reference pages for SCL are within the usual numeric sections and have an scl suffix.</p>
Executable program does not run.	Review the directions for installation and configuration and make any needed corrections.
Cannot port particular applications.	See the <i>Solaris Compatibility Libraries for Compaq Tru64 UNIX User's Guide</i> .

## Alternative Practices

Although this Best Practice is the recommended method for using Solaris Compatibility Libraries, more detailed information can be found in the *Solaris Compatibility Libraries for Compaq Tru64 UNIX User's Guide* at the following location:

<http://www.tru64unix.compaq.com/complibs/>

If your system does not meet the requirements described in *Is This Best Practice Right for You?*, you can port threaded applications to Tru64 UNIX using the POSIX Threads Library. Sun Microsystems Inc. supports the POSIX Threads Library, as well as the Solaris library.

## Comments and Questions

We value your comments and questions on the information in this document. Please mail your comments to us at this address:

`best_practices@zk3.dec.com`

## Legal Notice

Compaq and the Compaq logo Registered in U.S. Patent and Trademark Office. Tru64 is a trademark of Compaq Information Technologies Group, L.P. in the United States and other countries. UNIX is a trademark of The Open Group in the United States and other countries. All other product names mentioned herein may be trademarks of their respective companies.

Confidential computer software. Valid license from Compaq required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Compaq shall not be liable for technical or editorial errors or omissions contained herein. The information in this document is provided "as is" without warranty of any kind and is subject to change without notice. The warranties for Compaq products are set forth in the express limited warranty statements accompanying such products. Nothing herein should be construed as constituting an additional warranty.