
QuickTransit for Solaris/SPARC to Linux/ia64 v1.1 Release Notes

Legacy Server edition

QuickTransit for Solaris/SPARC to Linux/ia64 Release 1.1

For QuickTransit version QT 1.1 SW 1.1

Contact: <support-CL@transitive.com>

Copyright © 2007 Transitive Ltd

© 2007 Transitive Ltd. All rights reserved; provided portions may be copyright in third parties, as indicated elsewhere herein. No permission is granted to copy, distribute, or create derivative works from the contents of this electronic documentation in any manner, in whole or in part, without the prior written permission of Transitive Ltd.

This product is covered by one or more of the following US, UK and other Patents and/or published patent applications listed at <http://www.transitive.com/patents.htm>.

Transitive®, the Transitive Logo and QuickTransit are registered trademarks of Transitive Corporation and/or its affiliates.

Solaris and SPARC are registered trademarks of Sun Inc.

Linus Torvalds retains the copyright for Linux.

Red Hat is registered trademark of Red Hat Inc.

Itanium is a registered trademark of Intel Corporation.

FLEXlm is a trademark of Macrovision Corporation.

All other trademarks remain the property of their holders.

Overview

QuickTransit enables Linux/ia64 machines to run Solaris/SPARC applications alongside native Linux/ia64 applications. No modifications, recompiling or changes are needed to the Solaris/SPARC applications. The Solaris/SPARC applications simply need to be installed on or copied to a Linux/ia64 machine that has QuickTransit installed on it.

1 Release contents

This release contains the following files:

- Standalone release:

QuickTransit-SSLI-Legacy-1.1.tar.gz

Compressed tar file containing QuickTransit translation software. See the following table for details of contents.

SolarisWorld-1.1.linux.bin

A binary archive that contains the SolarisWorld files and directory structure.

- The compressed tar file `QuickTransit-SSLI-Legacy-1.1.tar.gz` contains the following items:

QuickTransit-SSLI-Legacy-1.1-0.bin

A binary archive that contains the QuickTransit translation software.

QuickTransit-Licensing-1.0-1.ia64.bin

A binary archive that contains the license server required for floating licenses.

QuickTransit-SSLI-Legacy-Release-Notes-1.1.pdf

QuickTransit-SSLI-Legacy-Release-Notes-1.1.txt

This document. Amendments to installation instructions and other information specific to the release version.

QuickTransit-SSLI-Legacy-Admin-Guide-1.1.pdf

QuickTransit-SSLI-Legacy-Admin-Guide-1.1.txt

The installation and administration guide.

QuickTransit-SSLI-Legacy-QuickStart-ReadMe-1.1.pdf

QuickTransit-SSLI-Legacy-QuickStart-ReadMe-1.1.txt

Quick start information.

SolarisWorld-Patch-5-1-1.linux.bin

A binary archive that contains updates and fixes to the SolarisWorld package. This fixes missing symbols in the supplied motif library.

SolarisWorld-Patch-6-1-1.linux.bin

A binary archive that contains updates and fixes to the SolarisWorld package. This adds `sshd` support to the Virtual Solaris Environment.

installer.sh

A script that installs, upgrades and uninstalls QuickTransit. It also installs a SolarisWorld from Transitive's own OpenSolaris based distribution.

default.config

An example configuration file for non-interactive installation. See the section "Using a configuration file to automate the Installation Script" in the QuickTransit Admin Guide.

upgrade.config

An example configuration file for non-interactive upgrade of an existing installation.

2 Release rationale

This is the 1.1 release of QuickTransit Solaris/SPARC to Linux/ia64.

3 Support and Contact Information

If you have a problem running a Solaris/SPARC application, please read section "Troubleshooting" in the QuickTransit Admin Guide. Please also read the section Chapter 6, *Limitations in this release* (page 3) of these release notes to see if there are any known limitations or problems that might apply.

If you require any further assistance, please e-mail <support-CL@transitive.com> with details of your problem, and any log files produced by QuickTransit.

4 System requirements

Please see the QuickTransit Admin Guide for details of hardware and software requirements. There are no additional system requirements specific to this release.

5 Installation instructions

Please see the QuickTransit Admin Guide for full installation and upgrade instructions.

5.1 Installing QuickTransit for the first time

For a quick-start installation, please refer to the `QuickTransit-SSLI-Legacy-QuickStart-ReadMe-1.1.txt` file, which details the steps needed to install QuickTransit using the supplied `default.config` answer file. This will install the latest version of QuickTransit, SolarisWorld and all related patches.

Alternatively, refer to 'QuickTransit Installation Reference' in the QuickTransit Admin Guide for information on installing QuickTransit via the interactive installation script.

**Note**

If you are installing via the interactive installer script, please ensure that you apply all supplied patches.

5.2 Upgrading from Beta releases is NOT supported

Upgrading to this release from a Beta release is not supported. If you have installed any QuickTransit Beta release then it must be completely removed from your system before installing this release.

5.3 Uninstalling previous releases

5.3.1 QuickTransit must be stopped before it is uninstalled

Before uninstalling QuickTransit, make sure that there are no QuickTransit processes still running by entering the following commands as root from Linux/ia64:

```
% /etc/init.d/QuickTransit stop
```

Make sure that the script used to uninstall QuickTransit is the one supplied with the version of QuickTransit that is being uninstalled.

5.3.2 Uninstalling may remove modified files

The `installer.sh` script gives the option to “Remove modified files and installed applications from SolarisWorld”. Even if you select to keep modified files, it is possible that certain modified files will be removed by the uninstaller. This problem is known to affect configuration files in the `/etc` directory.

If you have made important modifications to files in SolarisWorld that you need to keep, it is recommended that you make a backup copy of your SolarisWorld directory before it is uninstalled.

5.4 Changing from an evaluation license to a purchased license

If you have purchased QuickTransit, you will have received a FlexLM floating license file. This type of license requires a license server to be set up. This type of license is different from QuickTransit evaluation licenses, which do not require a license server. There are some additional steps necessary to install and use a purchased floating license. Please see the section “Changing from an evaluation license to a floating license” in the QuickTransit Admin Guide for details.

6 Limitations in this release

6.1 Packages missing from SolarisWorld

The Solaris/SPARC libraries, binaries and other infrastructure files that are provided by Transitive to create a SolarisWorld do not constitute a complete Solaris/SPARC system.

If you install the supplied SolarisWorld package and find that there are libraries or binaries missing that are required by your applications, then you will need to obtain and install the appropriate package.

Solaris/SPARC packages can be installed into SolarisWorld from a Solaris Express distribution. Please see the ‘Installation of Solaris Packages’ section of the QuickTransit Admin Guide.

Alternatively packages can be obtained from various freeware websites (for example, from a site such as <http://www.sunfreeware.com>) and installed by running the `pkgadd` tool from a Solaris/SPARC shell running within a Virtual Solaris Environment. Please see the QuickTransit Admin Guide for more information on installing applications in a Virtual Solaris Environment.

6.2 Limited support for raw device access

Raw (unbuffered) device access from the Virtual Solaris Environment is not currently supported. If your Solaris/SPARC application is normally configured to access raw devices, you may be able to configure it to access Linux/ia64 block devices instead by following the instructions below (as `root`).

1. Create an escape to the raw partition

```
% linksparc /dev/sda6
```

2. Determine the major and minor numbers of the Linux/ia64 device node

```
% ls -l /dev/sda6  
  
brw-rw---- 1 root disk 8, 6 Oct 16 11:02 /dev/sda6
```

In this example the major device number is 8 and the minor number is 6.

3. Use the major and minor device numbers to create an identical device in the SolarisWorld `/dev/bsync` directory. You may have to create the `bsync` directory first. For example, enter the following from a Linux/ia64 shell:

```
% mkdir <SOLARISWORLD_ROOT> /dev/bsync  
  
% mknod <SOLARISWORLD_ROOT> /dev/bsync/sda6 b 8 6
```

4. Now you should be able to see the devices from within a Virtual Solaris Environment as follows:

```
% runsparc  
  
$ ls -l /dev/sda6 /dev/bsync/sda6  
  
brw-r----- 1 root disk 8, 6 Oct 16 11:02 /dev/sda6  
  
brw-r--r-- 1 root sys 8, 6 Nov 21 16:27 /dev/bsync/sda6
```

This process will create two device nodes in the SolarisWorld, both pointing to the same physical partition. Both are block devices – they are not the same as a raw character device such as `/dev/rdisk/*` in Solaris. However, the two devices will differ in one important respect, the buffering of I/O. Access to the first device `/dev/sda6` may be buffered by Linux/ia64, with potential consequences for data consistency in the event of a machine crash. Access to the second device `/dev/bsync/sda6` will be specially handled by QuickTransit so that reads and writes are not buffered by the Linux/ia64 kernel.

6.3 Manually created escape symlinks

While it is possible to create escape symlinks manually, there are conditions under which it causes problems for QuickTransit. Therefore it is strongly recommended that escape symlinks are created with the `linksparc` command. If you manually create symlinks they must be created with one preceding “/” and no trailing slashes:

<code>ln -s /export /SolarisWorld/export</code>	correct
<code>ln -s //export /SolarisWorld/export</code>	incorrect
<code>ln -s /export/ /SolarisWorld/export</code>	incorrect

6.4 /proc tools

The following proc tools work but with the stated limitations:

Tool	Known limitations
<code>prstat</code>	SIZE and RSS fields include memory used by QuickTransit for internal use. The STATE field contains incorrect values.
<code>ps</code>	Processes started using <code>runsparc</code> have a PPID which is the process id of the Linux shell from which the <code>runsparc</code> command was entered. S, WCHAN, PRI and NICE fields contain incorrect values.
<code>pmap</code>	Some memory allocated for internal use by QuickTransit will appear in the process address map.
<code>pfiles</code>	Some files for internal use by QuickTransit will also appear in the output.
<code>pcred</code>	Setting credentials of another process will only work if the target process has root privileges. Additionally, credentials can be incorrectly reported.
<code>ptree</code>	The <code>-c</code> option is ignored.
<code>pldd</code>	Libraries provided by QuickTransit will appear in the output.
<code>pwait</code>	wait status will not be reported in verbose mode.

6.5 Debuggers

QuickTransit supports the debugging of applications within a Virtual Solaris Environment.

6.5.1 mdb

The Solaris/SPARC `mdb` tool has the following limitations in this release:

- Some memory allocated for internal use by QuickTransit will be visible using the `::objects` and `::mappings` commands.
- Watchpoints are not supported.
- In some cases, the threading model will be reported as “raw lwps” rather than “native threads”.
- `mdb` cannot be used to debug itself.
- It is not possible to follow either the parent or child process across a `vfork()`.
- After a single `step ::status` can give an incorrect “status::stopped...” message.

- The message "You've got symbols" may be reported when it would not be on Linux/ia64 systems.
- In some cases, if a breakpoint is set by function name the `::events` command will report it by address rather than by name.
- Occasionally, `mdb` may report "target is running" instead of "target has terminated".
- Occasionally, `mdb` may fail to stop at an `exec()` when instructed to.
- In some cases, `mdb` will give the location at which it is stopped as an absolute address, rather than `symbol+offset`.
- single stepping immediately after attaching to a running process may result in a crash.
- For multithreaded applications, `mdb` may identify a different representative thread from that which is seen on a Linux/ia64 system.

6.5.2 dbx

The Solaris/SPARC `dbx` tool has the following limitations in this release:

- Watchpoints are not supported.
- Checking for memory leaks (by using `check -all` for example) will cause a segmentation fault.
- `dbx` may behave incorrectly when sent signals from native Linux/ia64 processes.
- It is not possible to follow either the parent or child process across a `vfork()`.
- In some cases, `dbx` will stop twice at a breakpoint, when it should only stop once.
- Continuing execution after deleting a breakpoint may cause a crash.
- Occasionally `dbx` may fail with an error similar to "dbx:cannot read status for l@3 -- No such file or directory". In this case try restarting `dbx`.

6.5.3 gdb

The Solaris/SPARC version of `gdb` has the following limitations in this release:

- Watchpoints are not supported.
- When `gdb` is stopped at a signal (by using `handle <signal> stop`), the function and file in which the current execution location occurs are not identified.
- When `gdb` traps the throw or catch of a C++ exception, the address at which it stops will differ from that seen on a Linux/ia64 system.
- In some cases, extra warning messages are reported which are not seen on a Linux/ia64 system.

6.6 Java

QuickTransit supports Java version 1.3 and higher. Running Java versions 1.5 and 1.6 on QuickTransit require that the system kernel is recompiled to support 8k page sizes.

6.7 Limitations in other Solaris/SPARC system tools

The following general tools are known not to work at all in this release:

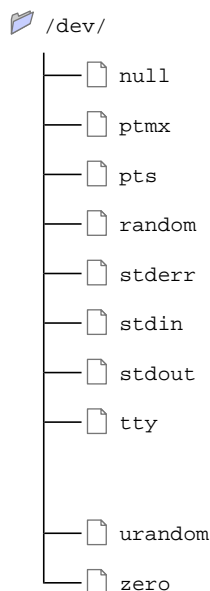
Tool
lastcomm
acctadm
fuser
prctl
ctstat

6.8 Devices supported by QuickTransit

This section lists the `/dev` device files that can be accessed.

6.8.1 Fully-Supported Devices

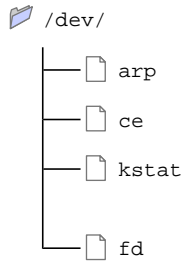
These devices are supported by using escapes from SolarisWorld to Linux/ia64, configured automatically at installation time.



tty devices escaped to Linux/ia64, tty devices are not full emulations of Solaris/SPARC terminals.

6.8.2 Solaris/SPARC Devices Emulated by QuickTransit

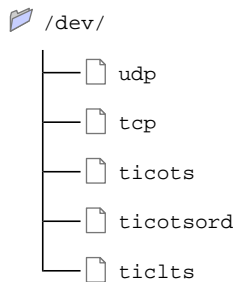
Devices listed in this section do not exist as actual files in the filesystem – instead they are emulated by QuickTransit at runtime. They can only be accessed by applications running within a Virtual Solaris Environment.



QuickTransit provides limited kstat information for modules “cpu”, “cpu_info” and “unix”.

6.8.3 Solaris/SPARC Transport Devices

These transport devices are all STREAMS-based devices, and hence are manipulated by getmsg and putmsg system calls. They are emulated at runtime and therefore do not exist as actual files in the SolarisWorld filesystem.

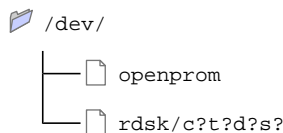


For supported STREAMS-based devices, not all of the ioctl STREAMS commands are fully supported. Unsupported or partially supported commands may give errors when they are passed to the translator. The following table identifies the extent to which different STREAMS commands are supported – fully, partially or not at all.

Fully Functioning Commands	Partially Supported Commands	Not Supported Commands
I_SRDOPT	I_PUSH	I_FLUSH
I_GRDOPT	I_POP	I_LINK
I_SWROPT	I_FIND	I_UNLINK
I_GWROPT	I_LOOK	I_RECVFD
I_PEEK	I_LIST	I_FDINSERT
I_NREAD	I_SETSIG	I_SENDFD
I_GETSIG	I_STR	I_PLINK
		I_PUNLINK
		I_ANCHOR
		I_FLUSHBAND
		I_CKBAND
		I_GETBAND
		I_ATMARK
		I_SETCLTIME
		I_GETCLTIME
		I_CANPUT
		I_SERROPT
		I_GERROPT
		I_ESETSIG
		I_EGETSIG
		__I_PUSH_NOCTTY
		_I_MUXID2FD
		_I_INSERT
		_I_REMOVE
		_I_GETPEERCRED
		_I_PLINK_LH

6.8.4 Unsupported Solaris/SPARC Devices

These devices exist natively on Solaris/SPARC, but are not supported at all by QuickTransit.



6.9 ypbind is automatically configured to use a ypserver list instead of network broadcasts

When the `installer.sh` script is run, it executes the `ypwhich` command to see if the Linux/ia64 NIS client is bound to a NIS server. If it is bound to a server then the address of the server is used to create a `ypservers` file in the SolarisWorld. When the Solaris/SPARC NIS client (`ypbind`) is started it then uses this `ypservers` file to locate a NIS server to use.

However, in some environments network broadcasts are used instead of server lists to locate NIS servers. The `installer.sh` script does not currently check which method is in use – if NIS is in use it assumes a server list is being used and configures the Solaris/SPARC `ypserver` file.

If you would prefer the Solaris/SPARC `ybind` daemon to use network broadcasts instead of server lists this must be configured manually. Please see the section ‘Configuring Solaris/SPARC `ybind`’ in the QuickTransit Admin Guide for details of how to do this.