7 September 2000

# HLA RTI 1.3
## Next Generation Implementation

# *RTI-NG 1.3v3.2 Release Notes*

The Next Generation RTI computer software and documentation has been developed under contract N61339-97-C-0073 in which the regulations from DFARS 252.227-7013 and 252.227-7014 are enforced. This material may be reproduced by or for the U.S. Government pursuant to the copyright license under these DFARS clauses.

Science Applications International Corporation
Technology Research Group
Distributed Computing Technology Division
5400 Shawnee Road, Suite 110
Alexandria, VA 22312

# 1 Introduction

The Run-Time Infrastructure Next Generation 1.3 (RTI-NG 1.3) is an implementation that corresponds to the High Level Architecture Interface Specification version 1.3 (http://hla.dmso.mil/tech/ifspec.html). The RTI provides a collection of common services used to support the modeling and simulation community. All of these services are accessed through a standard programming language API. The C++, Java, and Ada 95 programming languages are supported, as well as the CORBA Interface Definition Language. Information concerning the HLA and RTI can be found on the DMSO (Defense Modeling and Simulation Office) web site, http://hla.dmso.mil.

The RTI-NG 1.3 implementation has been developed under the sponsorship of the Defense Modeling and Simulation Office and contracted through the US Army Simulation, Training and Instrumentation Command (STRICOM). The development team has been led by the Sciences Application International Corporation (SAIC) Technology Research Group and supported by Virtual Technology Corporation, Object Sciences Corporation, Dynamic Animation Systems, Washington University in St. Louis, and University of California at Irvine.

## 1.1 About This Manual

This manual is intended for HLA federate developers and contains the latest information concerning the RTI Next Generation software release. Any known issues with the software will be identified in this document. These issues may include potential installation problems, software defects, or problems with adherence to the HLA Interface Specification.

Issues described within this document will either be common to all platforms or specific to a particular platform (i.e., machine architecture, operating system, operating system version, compiler, and compiler version combination). As additional releases to the RTI-NG software are distributed this manual will provide updates to any new features, as well as the known issues for that particular release.

Any issues encountered with the RTI-NG software or documentation should be brought to the attention of the RTI-NG support team so that any software defects or enhancement requests can be resolved and this document can be updated for other users of the RTI-NG software. A web based helpdesk system has been established to allow users to submit problem reports or enhancement requests. The location for the helpdesk system is http://helpdesk.dctd.saic.com, and helpdesk users are required to have a valid account with the DMSO Software Distribution Center.

## 1.2 Documentation

The RTI-NG documents are listed below. These documents can be downloaded individually from the DMSO Software Distribution Center (http://hla.dmso.mil/sdc). This documentation set includes:

***Installation Guide*** – This document explains how to install the RTI-NG software and verify that the installation is working properly. The installation verification process involves the building and execution of a simple federate application, helloWorld, which exercises a number of RTI services and exchanges information between two or more federates.

***Release Notes*** – This document contains any known issues for the latest release of the RTI-NG software. The known issues may include potential installation problems, software defects, or problems with adherence to the HLA Interface Specification.

***Programmer's Guide*** – This document describes how to develop federate applications using the RTI-NG software. Included is a description of all the RTI services and federate callbacks with additional guidance based on the behavior of the RTI-NG implementation.

# 2 Supported Platforms

The term "platform" is used to represent the combination of operating system, operating system version, machine architecture, compiler, compiler version, and compiler/link options used to build a particular RTI distribution. An RTI distribution is tested for only a particular platform configuration and may exhibit problems ranging from catastrophic failure to subtle behavioral differences when used under a different configuration. It is important that a user ensures that the RTI distribution matches the user platform configuration.

The list of supported platforms for the RTI-NG software is shown in the following table. Requests for additional platforms should be submitted through the DMSO RTI helpdesk.

Table 1. Supported Platforms

| **Operating System** | **Compiler Type** | **Compile/Build Type** |
|---|---|---|
| RedHat Linux 6.1 (Intel) | EGCS C++ 1.1.2 | Optimized, Multithreaded |
| SGI Irix 6.5.3 (MIPS) | MIPSpro C++ 7.2.1.3m | MIPS3, N32, Optimized |
| SGI Irix 6.5.6 (MIPS) | MIPSpro C++ 7.3.1.1m | MIPS4, N32, Optimized |
| Sun Solaris 2.6 (SPARC) | SunPro C++ 4.2 | Optimized, Multithreaded |
| Sun Solaris 7 (SPARC) | SunPro C++ 5.0 | Optimized, Multithreaded |
| Sun Solaris 7 (SPARC) | GNU C++ 2.95.2 | Optimized, Multithreaded |
| VxWorks 5.3.1 (PowerPC) | GreenHills C++ 1.8.9 | Optimized |
| Windows NT 4.0 SP6 | MS Visual C++ 6.0 SP3 | Debugged, Multithreaded, RTTI |
| Windows NT 4.0 SP6 | MS Visual C++ 6.0 SP3 | Optimized, Multithreaded, RTTI |
| Windows 98, 2nd Edition | MS Visual C++ 6.0 SP3 | Debugged, Multithreaded, RTTI |
| Windows 98, 2nd Edition | MS Visual C++ 6.0 SP3 | Optimized, Multithreaded, RTTI |

The "RTTI" acronym stands for run-time type checking and is used by the RTI software to ensure proper type-safety. The Windows releases provide a "debugged" version in order to be linked with debugged federate software because Microsoft uses a different underlying memory management scheme in debug mode.

The next Table contains the full set of compiler and link flags required for each platform.

| Operating System | Compiler Flags | Link Flags |
|---|---|---|
| RedHat Linux 6.1 (Intel) | none | -lRTI-NG -lfedtime -lpthread |
| SGI Irix 6.5.6 (MIPS4) | -n32<br>-D_POSIX_PTHREAD_SEMANTICS<br>-D_PTHREADS | -n32 -lRTI-NG –lfedtime<br>-lpthread |
| SGI Irix 6.5.3 (MIPS3) | -n32<br>-D_POSIX_PTHREAD_SEMANTICS<br>-D_PTHREADS | -n32 -lRTI-NG –lfedtime<br>-lpthread |
| Sun Solaris 2.6 (SPARC) Spro 4.2 | -features=castop -mt<br>-D__EXTENSIONS__<br>-D_POSIX_C_SOURCE=199506L<br>-D_POSIX_PTHREAD_SEMANTICS | -mt<br>-lRTI-NG -lgen -lposix4<br>-lpthread |
| Sun Solaris 7 (SPARC) Spro 5.0 | -mt -D__EXTENSIONS__<br>-D_POSIX_C_SOURCE=199506L<br>-D_POSIX_PTHREAD_SEMANTICS | -mt<br>-lRTI-NG -lgen -lposix4<br>-lpthread |
| Sun Solaris 7 (SPARC) gcc 2.95.2 | -D__EXTENSIONS__<br>-D_POSIX_C_SOURCE=199506L<br>-D_REENTRANT<br>-D_POSIX_PTHREAD_SEMANTICS | -lRTI-NG -lpthread |
| VxWorks 5.3.1 (PowerPC) | See subsequent VxWorks Section | See subsequent VxWorks Section |
| Windows NT 4.0 SP6 optimized | RTTI, multithreaded dll, RTI_USES_STD_FSTREAM if using namespace std for ostream. | Only linker flags automatically set by the RTTI and multithreaded flags. |
| Windows NT 4.0 SP6 debugged | RTTI, multithreaded debugged dll, RTI_USES_STD_FSTREAM if using namespace std for ostream | Only linker flags automatically set by the RTTI and multithreaded flags. |
| Windows 98, 2nd Edition optimized | RTTI, multithreaded dll, RTI_USES_STD_FSTREAM if using namespace std for ostream | Only linker flags automatically set by the RTTI and multithreaded flags. |
| Windows 98, 2nd Edition debugged | RTTI, multithreaded debugged dll, RTI_USES_STD_FSTREAM if using namespace std for ostream | Only linker flags automatically set by the RTTI and multithreaded flags. |

# 3 General Notes

This section contains information of general importance to using the RTI Next Generation software. These items are independent of the particular platform being used and are not necessarily defects with the software, but are useful for proper operation.

**RTI-NG Compatibility** – The RTI-NG implementation is not link-time or run-time compatible with non-NG RTI implementations. Link-time compatibility refers to the ability to have an existing federate application to utilize a different dynamically linked RTI. Due to a subtle difference in the RTI-NG interface it is necessary to recompile the federate software using the RTI-NG software. Run-time compatibility refers to federates utilizing different RTI implementations executing in a common federation execution. Due to differences in underlying algorithms and message formats RTI implementations from different vendors are not run-time compatible. Furthermore, changes between RTI-NG v1, v2, and v3 have resulted in differences that require all federates within a federation execution to utilize a common version. However, every effort has been made to allow interoperability between RTI-NG v3.1 and v3, though it is recommended that all federates within a simulation use the same version.

**RTI-NG Library Name Change**– For the v3 and later versions of the RTI-NG software, the file name of the delivered code library has changed from previous versions. Previous versions of RTI-NG have provided libRTI as the library name. Starting with v3, this name changes to libRTI-NG. In addition, instead of one large library that contains all the code, the v3 and later RTI-NG distributions are released as several independent libraries with the main libRTI-NG providing all the dependencies on the sublibraries. In effect for the federate developer, the only change required is to link against libRTI-NG instead of libRTI.

**RTI Console Tool Forced Removal Requires MOM** – The use of the RTIConsole Tool requires that the federation use MOM. Use of the tool is inconsistent with a failure to use MOM. The behavior is undefined if utilized in a federation that does not have MOM enabled.

**ACE/TAO** – The RTI-NG software internally uses the Adaptive Communication Environment (ACE) and The ACE ORB (TAO) as a low-level communication framework. Washington University at St. Louis has developed these software packages under the direction of Dr. Douglas Schmidt. Information concerning ACE and TAO can be found at http://www.cs.wustl.edu/~schmidt. In the event that a federate application is attempting to use either ACE or TAO directly, please contact the RTI helpdesk (http://helpdesk.dctd.saic.com) to ensure both the application and the RTI software are using compatible versions. The ACE/TAO libraries are provided with the RTI-NG release; they are located in the same directory as the RTI-NG libraries.

**Standard C++ Library** – Until all of the compiler vendors properly support the ANSI/ISO C++ standard and deprecate the older constructs there are issues when devel-

oping software to work under the standard and non-standard environments. One of the most common issues is that the standard library uses name spaces for compiler provided standard definitions, although most compilers still support the un-scoped names as well. For example, an application can choose to include fstream.h and use cout for streaming to standard output, or the application can use the standard convention to include fstream with a using declaration or explicitly scoping with std::cout.

In the case of the RTI header files the fstream class was exposed to the federate application in the 1.3 Interface Specification. In order to support the standard and non-standard conventions, the federate software that includes the RTI header files must be compiled with the directive RTI_USES_STD_FSTREAM when utilizing the newer C++ standard of using namespaces to access stream functions.

**Relevance Advisories** – The RTI relevance advisories are necessary to inform the federate that a particular object or interaction class, or a particular object attribute instance is relevant to the federation execution. These items are relevant if there is another federate that is interested in the data that may be exchanged by that item. Relevance advisories will inform a federate whether it should (or should not) continue to register objects or send interactions of a particular type, or perform updates of a particular object attribute instance.

In the RTI-NG implementation the relevance advisories operate using a heartbeat mechanism to improve scalability. The RID file defines parameters used to control the behavior of how often the heartbeat occurs and how long the RTI will wait to determine when an item is marked not relevant. The heartbeat mechanism results in the relevance advisories being asynchronous to the actual federate RTI invocations that define whether something is or is not relevant. This can be disturbing when looking for immediate causal relationships with these RTI services.

Due to the performance overhead of the relevance advisories it is recommended that the RID parameters ClassRelevanceAdvisorySwitch, InteractionRelevanceAdvisorySwitch, and AttributeRelevanceAdvisorySwitch be disabled when the federation does not utilize these services. Disabling these advisories across the federation execution will allow the RTI to avoid performing calculations and distributing state that are required to properly support the {Start,Stop}ObjectRegistration, Turn{On,Off}Interactions, and TurnUpdates{On,Off}ForObjectInstance callback services.

**Network Requirements** – The distributed communication protocols used for the RTI-NG implementation are based on Internet Protocol (IP), specifically TCP (Transmission Control Protocol) for “Reliable” transport and UDP (User Datagram Protocol) multicast for “Best Effort” transport. The network used to connect federate applications operating within a federation execution must support IP based communication. A multiple Local Area Network (LAN) configuration is supported even when UDP multicast is not routed between these networks (although only reliable transport for attributes and interactions can be used). The RID file contains parameters to support different network configurations, including non-multicast discovery and non-multicast WAN environments. Please consult the RTI-NG Installation Guide for specific information.

**RTIExec No Longer Accepts Commands –** Starting with the RTI-NG v3, instead of accepting commands from the rtiexec console window, a separate tool for controlling the removal of federates and federations is provided called the rtiConsole tool. This tool provides a number of benefits to the federate developers and users. The tool can be run in a distributed fashion from any computer on the network. This tool connects to the rtiexec and can be used to monitor and control an on-going simulation. Refer to the installation guide for detailed usage information for the rtiConsole.

# 4 Known Issues

This release of the Next Generation RTI has undergone significant internal and external testing to determine software defects and differences with the existing DMSO RTI 1.3 implementation. This RTI implementation has been officially verified through the DMSO RTI Verification process and periodic re-verification will be performed. The testing feedback has gone back into improving the correctness and robustness of the software, although some items were not detected until the software was being prepared for distribution. The software defects listed below apply to all of the platform configurations, and are expected to be resolved with the next release.

It is also noted that the RTI-NG development and testing environment is not capable of reproducing all environments and conditions in which the RTI is expected to operate. Please utilize the RTI helpdesk when you encounter problems using the software so that defect resolutions and enhancements can be incorporated in subsequent releases.

**Reflects of Transferred Deleted Attributes** – If a federate deletes a particular object instance and that federate has a pending reflect of an attribute belonging to the object, but owned by another federate, that pending reflect will not be removed.

**Object Registration Time** – The time required to register an object scales linearly with the number of attributes.

**Cannot Specify Multicast Interface for Rtiexec** – There is currently no way to specify the network interface to use for the rtiexec. If use of a NIC other than the one specified by the multicast route is desired, use the -endpoint argument to rtiexec and specify RtiExecutiveEndpoint in the RID file.

# 5 Resolution of Previously Reported Issues

## 5.1 RTI-NG 1.3v3 Resolved Issues

**MOM Counting Inconsistency** – Previously updateAttributeValues with empty attribute handle value pair sets, i.e, an attribute handle value pair set with no handles or values, were sent but discarded by the receiving federate. This caused confusion in MOM counting in which updates were counted as sent but would not be received by any federate due to being empty. The empty updateAttributeValues are no longer sent, correcting this problem.

**RTI Abort in Ownership Management** – Certain sequences of ownership management activities could cause the RTI to abort due to an unexpected exception being thrown. This has been corrected.

**Federation Synchronization** – Federation synchronization has been fixed to recognize forced removal of a federate.

**RTI Version Compatibility** – Introduced checking for RTI version compatibility when a federate creates and/or joins a federation execution. Since it is not possible to deduce the version for federates using v3 or older, a warning is printed out. The federate is allowed to continue with its activity.

**Memory Growth with TSO Updates** – Federates, which received TSO updates, could experience slow but unbounded memory growth due to improper caching of event retraction handles. This problem could also cause event retractions not to be delivered under some certain rare conditions. The memory growth problem has been corrected. Event retraction for constrained federates is also corrected. Event retractions for unconstrained federates may not be delivered correctly if the RID parameter EventRetractionHandleCacheOptions.MinimumCacheSizeBeforePerformingPurge is not large enough, but this is extremely unlikely unless the receiving federate does not tick for long periods of time.

**Missing timeRegulationEnabled() Callback** – When the last time regulating federate in a federation resigned, it was possible that the next federate which tried to become time regulating would never get the timeRegulationEnabled() callback. This has been corrected.

**Relevance Advisories After Save and Restore** – During and immediately after a save or restore, it was possible for federates to receive incorrect relevance advisories. This problem has been corrected.

**Memory Usage** – Memory usage per object instance and per attribute instance has decreased significantly.

**Multicast.Interface RID Parameter Ignored** – The parameter Multicast.Interface in the RID file was unintentionally ignored in v3. This parameter now works.

**RtiConsole "-endpoint" Does Not Work** – The rtiConsole tool could not connect to a rtiexec, which was started with the –endpoint <host:port> option using the rtiConsole -endpoint <host:port> option. This now works using a new -rtiexecEndpoint <host:port> option with the rtiConsole. The old option has been removed.

**Rtiexec "-endpoint" Broken on IRIX MIPS 3** – Previously the rtiexec for the IRIX-6.5-MIPSpro-7.2-mips3 could not launch the fedex process if started with the –endpoint <host:port> option and would core dump. This problem has been corrected.

**Rtiexec Crashes When Bad "-endpoint" Specified** – The rtiexec could core dump if a bad or improper –endpoint <host:port> was specified. The rtiexec now prints an error message and exits.

## 5.2 RTI-NG 1.3v2 Resolved Issues

**Hostname Dependency** – Previous implementations of RTI-NG relied on proper host-name configuration for each computing platform within the federation execution. This caused a number of difficulties with machines that did not have hostname, did not have fully qualified hostnames, or had other configuration anomalies. We have modified the underlying communication framework for RTI-NG to only utilize IP addresses

**RTI Executive Diagnostic Messages** – When a federate abruptly left a federation execution without the nominal resign invocation the RTI Executive process could become inundated with connection error messages. This undesirable behavior has been resolved.

**Forced Removal of Time Managed Federates** – In the previous implementation a time managed federate that was forced to resign using the MOM service or the RTI Executive console the time advancement algorithm would halt. The current implementation will wait a configurable amount of time after a forced resign and assume that all time-stamped messages have been delivered to allow the synchronization algorithm to proceed.

**Advisory Scalability** – A problem was detected with multiple federates simultaneously registering a large number of objects with the RTI-NG 1.3v2 implementation. The algorithm supporting the relevance advisories was improved to alleviate this issue.

**Consecutive Join/Resign Cycles** – The previous version would sometimes exhibit problems when federates would rapidly join and resign from a federation. Since these issues are very sensitive to timing conditions it is difficult to predict with certainty that all problems related to these operating conditions have been resolved, although tests constructed to exhibit these errors have been run successfully with the latest release.

## 5.3 RTI-NG 1.3v1 Resolved Issues

The items listed below are the list of user reported issues from RTI-NG 1.3v1 that have been fixed in this release.

**The "Objects Reflected" Count from the MOM Erroneously Includes the Objects Owned in the Count** – The MOM previously included the locally owned objects in its objects reflected count.

**The "Update Sent" Count from the MOM Erroneously Includes the Objects Owned in the Count** – The MOM previously included the locally owned objects in its updates sent count.

**An Intermittent Error in Receive Interaction Printed when using MOM to Invoke Methods** – Under certain circumstances, the use of MOM to invoke methods on a remote federate resulted in an error message being printed to the console.

**During the Remove Object Callback, Internal RTI State Deleted Before Callback** – This particular error did not allow the federate to look up things like the object name during the callback because the internal RTI data about the object had already been deleted before the callback.

**Next Event Request Call Could Result in a Premature Time Advance Grant Callback** – This behavior could also cause reliable traffic to be lost since the time on the packet would be less than the current time of the Time Advance Grant, causing the RTI to discard the packet. This error could also result in out of order delivery of Time Stamp Ordered packets.

**Join and Resign reportServiceInvocations are Not Sent – The Join and Resign calls were not causing report service invocations to be sent by MOM.**

# 6 Microsoft Windows 98/NT/2000

## 6.1 System Requirements

The RTI-NG 1.3 implementation supports Windows 98, 2nd Edition, version 4.0 of the Microsoft Windows NT, and Windows 2000 operating systems using the Microsoft Visual C++ compiler version 6.0 running on the Intel x86 architecture. Currently, Windows 95 is not supported nor tested, although this release may work on that OS version. The new Windows 2000 service pack 1 has not been applied or tested with this release.

### 6.1.1 Required Patches

The NT operating system requires Service Pack 5 and the compiler requires service pack 3 (i.e., Windows NT 4.0 SP5 and Visual C++ 6.0 SP3). Windows 98 requires the second edition version and the Visual C++ 6.0 compiler requires service pack 3.

## 6.2 Platform Specific Known Issues

**Config tool** – The config tool needs to be run from the Winnt-4.0-VC6 directory in the following manner: etc\print_sys_info.bat > sys_info.txt. Then, any text editor can be used to examine the contents of sys_info.txt. This file is described in the Programmer's Guide, and it includes important system information that can be used to diagnose runtime problems.

**Standard FSTREAM** – As mentioned above in section 2, the Windows Visual C++ compiler requires the use of the compiler flag RTI_USES_STD_FSTREAM if using the new std::stream form of the iostream library. Only use this flag if you are using the new standard C++ mechanism of namespaces to access the stream functions and operators.

**Compiler Flags and Options** – All federate applications that are built using the RTI on Windows 98/NT/2000 must have the RTTI compile flag set, as well as use the Multi-threaded DLL option in the C/C++ code generation compiler options section.

**Setting the Path to the Fedex in the RID File on 98/NT Fails** – Setting the path to the fedex on NT fails to correctly launch the fedex when using Windows style path separators. Instead, UNIX style path separators should be used. For example, c:\RTI would be c:/RTI in the RID file. Another workaround is to change the working directory to the directory where the rtiexec is located and run the rtiexec from there with the path to the fedex commented out in the RID file.

**Absolute Path to fedex in RID** – Under NT, setting a path for the fedex in the RID file requires the use of UNIX style slahes rather than NT style, so the c:\rti would be c:/rti in the RID file. Another workaround is to simply start the rtiexec from directory in which the fedex executable is located, or give the relative path to the rtiexec when starting it in a dos shell in any other directory.

**Performance anomaly for large attributes or parameters (~1000 bytes)** – On Windows NT, a performance anomaly has been found for attribute or parameter sizes around 1000 bytes. Changing the buffering RID parameters may improve this situation, but the resolution of this problem will not exist until the next release.

# 7 RedHat Linux

## 7.1 System Requirements

The RTI-NG 1.3 implementation supports version 6.1 of Red Hat Linux (kernel 2.2) using version 1.1.2 of the egcs C++ compiler running on the Intel x86 architecture. This configuration represents the standard releases of the operating system and compiler without any updates, patches, or extensions as found at the Red Hat ftp site: ftp://ftp.redhat.com. We have done limited testing on both Red Hat 6.0 and Red Hat 6.2 stock installations and both appear to work as well.

### 7.1.1 Required Platform Patches

None.

## 7.2 Platform Specific Known Issues

**Debugger Crashes** – The gdb debugger that is distributed with the Red Hat Linux 6.0 distribution (gdb-4.17.0.11) often core dumps when running RTI applications. Upgrading to gdb-4.18 (available from ftp://ftp.gnu.org/pub/gnu/gdb) fixes this problem.

# 8 SGI IRIX

## 8.1 System Requirements

The RTI-NG 1.3 implementation supports two SGI IRIX operating system versions and compiler flags. Currently, the primary combination is IRIX 6.5.6, using SGI MIPSpro C++ compiler version 7.3.1.1 running on the MIPS-4 ISA (Instruction Set Architecture) with the N32 ABI (Application Binary Interface). RTI-NG 1.3 also supports IRIX 6.5.3 using compiler MIPSpro 7.2.1.3 running on the MIPS-3 ISA and N32. Due to bloated library size and performance limitations with the IRIX 6.5.3 MIPSpro 7.2.1.3 MIPS 3 release, it is recommended that developers use the IRIX 6.5.6 MIPSpro 7.3.1.1 MIPS 4 release if possible. Note: the SGI release notes for the MIPSpro 7.3.1.1 compiler state that it is "fully compatible" with the 7.2.1 compiler, except for linking WHIRL object files. Thus the MIPS 4 release should link successfully using the 7.2.1 compiler, though this cannot be guaranteed to work for every federate. The MIPS 4 release also requires that machines have MIPS R5000 chips or newer.

### 8.1.1 Required Patches

The minor number on the operating system and compiler indicates the patch scheme being used by SGI. For example, the operating system version 6.5.3 indicates a patch level 3 to the standard 6.5 version of the operating system. Provided that the target system is using the 6.5.3 operating system and the 7.2.1.3 C++ compiler the appropriate patches will have been applied to the machine.

## 8.2 Platform Specific Known Issues

**MOM True/False Strings** – Where MOM parameters should report the string "True" or "False", the IRIX version is supplying the string "1" and "0", respectively.

**Invalid FED File** – If an invalid FED file is provided during the CreateFederation service it may result in a core dump after the exception is raised.

**Library Size** – Due to a problem with the handling of template code in the MIPSpro 7.2 compiler/linker the RTI libraries suffer from bloated size. This issue has been improved with the newer 7.3 compiler.

# 9 Sun Solaris

## 9.1 System Requirements

The RTI-NG 1.3 implementation supports Sun Solaris version 2.7 the Sun C++ compiler version 4.2. In addition, the RTI-NG 1.3v3.1 supports Solaris 7 using either the Sun C++ compiler version 5.0 with a special private patch or the GNU compiler version 2.95.2 running on the SPARC architecture.

### 9.1.1 Required Solaris 2.6 Patches

The following patches to the operating system and compiler are required for proper operation. The latest patch version should be installed in the case that one of the patches listed below has been superceded.

104631-07 SPARCompiler C++ 4.2: C++ 4.2 patch for Solaris 2.x

104668-09 SPARCompiler C 4.2: C patch for Solaris 2.x

105490-07 SunOS 5.6: linker patch

105568-13 SunOS 5.6: /usr/lib/libthread.so.1 patch

105591-06 SunOS 5.6: Shared library patch for C++

106125-08 SunOS 5.6: Patch for patchadd and patchrm

The following patches apply to the operating system and have been used during development and testing. It is unclear whether they are necessary for proper operation of the RTI, but in case there are problems it may be useful to have them installed.

105621-17(09) SunOS 5.6: libbsm and cron patch

105181-16(10) SunOS 5.6: kernel update patch

105210-24 SunOS 5.6: libc & watchmalloc patch

105222-03 SunOS 5.6: sbus driver patch

105223-05 SunOS 5.6: pln/soc drivers & ssafirmware patch

105377-03 SunOS 5.6: BCP patch

105379-05 SunOS 5.6: /kernel/misc/nfssrv patch

105397-02 SunOS 5.6: /usr/sbin/passmgmt patch

105401-20(16) SunOS 5.6: libnsl and NIS+ commands patch

105426-01 SunOS 5.6: /usr/lib/libtnfprobe.so.1 patch

105528-01 SunOS 5.6: /kernel/drv/be patch

105529-08 SunOS 5.6: /kernel/drv/tcp patch

105552-02 SunOS 5.6: /usr/sbin/rpc.nisd_resolv patch

105580-13 SunOS 5.6: /kernel/drv/glm patch

105600-10 SunOS 5.6: /kernel/drv/isp patch

105604-08 SunOS 5.6: ebus/pci/rootnex driver patch

105742-05 SunOS 5.6: /kernel/drv/le patch

105743-01 SunOS 5.6: /usr/lib/libxfn.so.2 patch

105755-07 SunOS 5.6: libresolv, in.named, named-xfer, nslookup, nstest patch

105778-01 SunOS 5.6: /kernel/fs/specfs patch

105780-04 SunOS 5.6: /kernel/fs/fifofs patch

105786-08 SunOS 5.6: /kernel/drv/ip patch

105795-05 SunOS 5.6: /kernel/drv/hme patch

105797-06 SunOS 5.6: /kernel/drv/sd patch

105836-03 SunOS 5.6: /kernel/drv/qe patch

105924-10 SunOS 5.6: kbd, se and zs drivers patch

106044-01 SunOS 5.6: /usr/lib/nss_nisplus.so.1 patch

106168-02 SunOS 5.6: dma driver patch

106169-02 SunOS 5.6: sbusmem driver patch

106170-03 SunOS 5.6: /kernel/drv/esp patch

106171-01 SunOS 5.6: /kernel/drv/lebuffer patch

106172-04 SunOS 5.6: /kernel/drv/fas patch

106173-03 SunOS 5.6: /kernel/misc/scsi patch

106183-04 SunOS 5.6: cfgadm utility & libraries

106193-03 SunOS 5.6: y2000 sysid unzip patch

106216-02 SunOS 5.6: /platform/sun4u/kernel/drv/envctrl patch

106323-01 SunOS 5.6: /etc/inet/services patch

### 9.1.2 Solaris 7 patch

The SunPro 5.0 compiler version of the RTI-NG requires the use of a special patch that was in beta testing at the time of this writing. The patch fixes a compiler bug in which exceptions thrown will cause the executable to core dump when thrown. The patch required was labeled: 107311-10 temporary patch.

## 9.2 Platform Specific Known Issues

None.

# 10 Wind River Systems VxWorks

## 10.1 System Requirements

The RTI-NG 1.3 implementation supports version 5.3.1 of the Wind River Systems VxWorks operating system using the Green Hills C++ compiler version 1.8.9 running on a large number of Wind Rivers supported VME processor cards. Our testing has involved Motorola 1604 and 2604 boards with at least 32 Mbytes of memory.

### 10.1.1 Required Patches

Tornado PowerPC Cumulative Patch 1

## 10.2 Platform Specific Known Issues

**No Rtiexec or Fedex** – The "rtiexec" and "fedex" applications are not distributed for VxWorks. Users must run these applications on one of the other platforms supported by the RTI.

**Limited Support for Concurrent RTI Tasks** – The RTI has not been tested in an environment where several independent instances of the RTI are used concurrently on the same VxWorks target. Users wishing to use the RTI with several concurrent tasks are advised to link all of the tasks' object code into a single binary image with a single entry point that instantiates the RTI static objects (as described in the Installation Guide) and launches all of the concurrent tasks. This is roughly analogous to having a single multi-threaded process on a conventional operating system. For further assistance, please contact the RTI-NG support staff.

**Target Reboot Required Between Successive RTI Runs** – The VxWorks RTI requires that the VxWorks target be rebooted in between successive runs of the RTI, where a "run" is defined as the lifetime of the RTI static objects (as described in the Installation Guide) on the program stack.

**RTI Library Size** – Although the file size of the RTI library is over 53 Mbytes, much of this size is attributable to relocation symbols that do not contribute to the target memory footprint of an application. The *sizeppc* and *gsize* programs can be used to determine the static target memory footprint of an executable or library. The total static footprint of the RTI library and all of its dependencies (libACE, libTAO, etc.) is approximately 16 Mbytes.