

Changes in C++ TargetRTS

from RoseRT to RSARTE

Elena Volkova, Senior Software Engineer
IBM

Revision history

Date	Version	Description	Author
2014.May.26	1.0	Initial Release	Elena Volkova
2014.June.03	1.1	Added information about Connexis stack size settings change	Elena Volkova
2018.March.19	2.0	Added information about all recent changes up to RSARTE 10.2 2018.09	Elena Volkova
2018.July.12	2.1	Updates for RSARTE 10.2 2018.27 – support for MinGW gcc 8.1.0 added	Mattias Mohlin
2018.August.20	2.2	Updates for RSARTE 10.2 2018.33 – support for Visual Studio 2017 (15.0) added	Mattias Mohlin

This document describes all changes made in the C++ TargetRTS for Rational Software Architect Real-Time Edition (RSARTE). All changes compared to the version that is included in RoseRT are covered. The purpose of the document is to help users incorporate TargetRTS changes into their own versions of the TargetRTS.

Table of contents

Support invoke() without reply()....3
Defer invoked replies....3
Remove usage of obsolete RTRUNTIMEBC macro....3
Remove obsolete target configurations....4
Added support for 64-bit Visual Studio and some other new target configurations....6
Updated Copyright texts....6
Fix problem in main.mk to make it work for MinGW....6
Update stack size settings for Connexis on SLED10x64T....7
Better error reporting in Perl scripts....7
Added support for Quantify and Pure Coverage in rtlink.pl....7
Improved configuration of paths for GNU compiler in setup.pl....7
Added class-scoped state entry operation rtgStateEntry....7
Passing information about initiator to some RTActor functions....8
Added support for unsigned integer basic data types to ascii encoding/decoding and logging functions....9
Added more accurate error message about no free ports triggered by FrameImport....9
Added support for IPv6 protocol....9
Print information about port index when receiving unexpected message to port....9
Improved TargetRTS multi-threaded mode (when USE_THREADS is defined)....10
Added support for 64-bit MinGW GCC 8.1.0....10
Added support for Visual Studio 2017 (15.0)....10

Support invoke() without reply()

Previously the TargetRTS has required that when you invoke() a signal, then the receiver needs to do a reply(). When the callee does not have any data to pass back to the caller, it is unnecessary to require an explicit reply(). Sometimes the caller just wants to ensure that a signal is synchronously sent without getting any return value. It's analogous to calling a void function in C++.

The TargetRTS has therefore been extended so that it now is optional to provide a reply buffer when calling invoke().

These files were updated:

```
<TargetRTS>/include/RTOutSignal.h
<TargetRTS>/include/RTOutSignal.inl
<TargetRTS>/include/RTProtocol.h
<TargetRTS>/src/RTProtocol/invoke.cc
<TargetRTS>/src/RTProtocol/invokeAt.cc
<TargetRTS>/src/RTProtocol/invokeOne.cc
<TargetRTS>/include/RTSymmetricSignal.h
<TargetRTS>/include/RTSymmetricSignal.inl
```

Defer invoked replies

A new function RTActor::sendCopyToMe() has been added to allow a capsule to send a copy of a message to itself.

Example:

```
RTMessage reply;
myport.myevent().invoke(&reply);
sendCopyToMe(&reply);
```

More generally this function can be used whenever a message cannot be fully handled in a single transition.

These files were updated:

```
<TargetRTS>/include/RTActor.h
<TargetRTS>/include/RTActor.inl
<TargetRTS>/include/RTController.h
<TargetRTS>/include/RTController.inl
```

Remove usage of obsolete RTRUNTIMEBC macro

Removed conditional sections #if RTRUNTIMEBC ... #endif from the following files:

```
<TargetRTS>/include/RTConfig.h
<TargetRTS>/include/RTMessage.h
<TargetRTS>/include/RTMessage.inl
<TargetRTS>/include/RTProtocol.h
<TargetRTS>/include/RTProtocol.inl
<TargetRTS>/include/RTProtocolDescriptor.h
<TargetRTS>/src/MANIFEST.cpp
<TargetRTS>/src/RTActor/control.cc
<TargetRTS>/src/RTException/descriptor.cc
<TargetRTS>/src/RTExternal/descriptor.cc
<TargetRTS>/src/RTFrame/descriptor.cc
<TargetRTS>/src/RTLog/descriptor.cc
```

```
<TargetRTS>/src/RTRootProtocol/descriptor.cc
<TargetRTS>/src/RTTiming/descriptor.cc
<TargetRTS>/target/SLED10T/RTTarget.h
<TargetRTS>/target/SLED10x64T/RTTarget.h
<TargetRTS>/target/VxWorks60T/RTTarget.h
<TargetRTS>/target/VxWorks64RTPT/rttarget.h
```

Remove obsolete target configurations

Removed the following folders from <TargetRTS>/config/ :

```
AIX4S.ppc-gnu-2.8.1
AIX4T.ppc-gnu-2.8.1
CHORUS40T.ppc603-egcs-2.91.66
IRIX6S.r4400-gnu-2.8.1
IRIX6T.r4400-gnu-2.8.1
LYNX30S.ppc-cygnus-2.7-97r1
LYNX30S.x86-cygnus-2.7-97r1
LYNX30T.ppc-cygnus-2.7-97r1
LYNX30T.x86-cygnus-2.7-97r1
LYNX31S.ppc-gnupro-2.9-98r2
LYNX31T.ppc-gnupro-2.9-98r2
NT40CygwinT.x86-cygwin-gnu-2.95.3-5
NT40CygwinT.x86-cygwin-gnu-3.2
NT40CygwinT.x86-cygwin-gnu-3.4.4-3
NT40CygwinT.x86-cygwin-gnu-4.5.3
NUCLEUS11T.ppc860-Diab-4.2b
NUCLEUS11T.x86-VisualC++-6.0
PSOS25T.ppc603-Diab-4.2b
PSOS25T.ppc860-Diab-4.2b
QNX4S.x86-WC++-10.6
REDHAT61S.x86-egcs-2.91.66
REDHAT61T.x86-egcs-2.91.66
REDHAT73MT.x86-gnu-3.2
REDHAT73T.x86-gnu-3.2
REDHAT80MT.x86-gnu-3.2
REDHAT80T.x86-gnu-3.2
SOLARIS10SPARC64T.sparc64-gnu-3.4.6
SOLARIS10T.sparc-gnu-3.4.6
SUN5MT.sparc-SunC++-5.0
SUN5S.sparc-SunC++-5.0
SUN5S.sparc-gnu-2.7.2.3
SUN5S.sparc-gnu-2.8.1
SUN5S.sparc-gnu-2.95.1
SUN5T.sparc-SunC++-5.0
SUN5T.sparc-SunC++-5.3
SUN5T.sparc-gnu-2.7.2.3
SUN5T.sparc-gnu-2.8.1
SUN5T.sparc-gnu-2.95.1
SUN5T.sparc-gnu-3.0.4
TORNADO101T.i960-cygnus-2.7.2-960126
TORNADO101T.m68040-cygnus-2.7.2-960126
TORNADO101T.ppc-cygnus-2.7.2-960126
TORNADO101T.ppc860-cygnus-2.7.2-960126
TORNADO101T.x86-cygnus-2.7.2-960126
TORNADO22T.ppc603-Diab-5.0.1
TORNADO22T.ppc603-gnu-2.96
TORNADO22T.simnt-gnu-2.96
TORNADO22T.simsparcsolaris-gnu-2.96
TORNADO2T.m68040-cygnus-2.7.2-960126
TORNADO2T.ppc-cygnus-2.7.2-960126
TORNADO2T.ppc603-GreenVX-1.8.9
TORNADO2T.ppc603-GreenVX-2.0
TORNADO2T.ppc860-cygnus-2.7.2-960126
```

```
TORNADO2T.simpc-egcs-2.90.29
TORNADO2T.simso-cygnus-2.7.2-960126
TORNADO2T.x86-cygnus-2.7.2-960126
UNIXWARE701S.x86-SDK-3.0
UNIXWARE701T.x86-SDK-3.0
VRTX4T.ppc603-Microtec-1.3C
VRTX4T.ppc603-Microtec-1.4
VxWorksAE10T.ppc603-gnu-2.96
VxWorksAE10T.simnt-gnu-2.96
VxWorksAE10T.simsparcsolaris-gnu-2.96
WINCE300T.arm-eMVisualC++-3.0
WINCE300T.mips-eMVisualC++-3.0
WINCE300T.sh3-eMVisualC++-3.0
```

Removed the following folders from <TargetRTS>/src/target/ :

```
AIX4
CLASSIX31
IRIX6
LYNX2
NT40Cygwin
NUCLEUS
PSOS2
QNX4
SOLARIS10
SOLARIS10SPARC64
SUN5
TORNADO1
UNIXWARE212
VRTX3
WINCE
```

Removed the following folders from <TargetRTS>/target/ :

```
AIX4S
AIX4T
CHORUS40T
IRIX6S
IRIX6T
LYNX30S
LYNX30T
LYNX31S
LYNX31T
NT40CygwinT
NUCLEUS11T
PSOS25T
QNX4S
REDHAT61S
REDHAT61T
REDHAT73T
REDHAT80T
SOLARIS10SPARC64T
SOLARIS10T
SUN5MT
SUN5S
SUN5T
TORNADO101T
TORNADO22T
TORNADO2T
UNIXWARE701S
UNIXWARE701T
```

```
VRTX4T
VxWorksAE10T
WINCE300T
```

Removed the following folders from <TargetRTS>/lib/ :

```
CHORUS40T.ppc603-egcs-2.91.66
LYNX30T.ppc-cygnus-2.7-97r1
LYNX30T.x86-cygnus-2.7-97r1
LYNX31T.ppc-gnupro-2.9-98r2
NT40CygwinT.x86-cygwin-gnu-3.2
NT40CygwinT.x86-cygwin-gnu-3.4.4-3
NT40CygwinT.x86-cygwin-gnu-4.5.3
NT40MT.x86-VisualC++-6.0
REDHAT73MT.x86-gnu-3.2
REDHAT73T.x86-gnu-3.2
REDHAT80MT.x86-gnu-3.2
REDHAT80T.x86-gnu-3.2
SOLARIS10SPARC64T.sparc64-gnu-3.4.6
SOLARIS10T.sparc-gnu-3.4.6
SUN5MT.sparc-SunC++-5.0
SUN5T.sparc-SunC++-5.0
SUN5T.sparc-SunC++-5.3
SUN5T.sparc-gnu-2.8.1
SUN5T.sparc-gnu-2.95.1
SUN5T.sparc-gnu-3.0.4
TORNADO22T.ppc603-gnu-2.96
TORNADO22T.simsparcsolaris-gnu-2.96
TORNADO2T.m68040-cygnus-2.7.2-960126
TORNADO2T.ppc-cygnus-2.7.2-960126
TORNADO2T.ppc860-cygnus-2.7.2-960126
TORNADO2T.x86-cygnus-2.7.2-960126
WINCE300T.sh3-eMVisualC++-3.0
```

Added support for 64-bit Visual Studio and some other new target configurations

The TargetRTS now contain config and libset files for all cases of 64-bit compilation: native (when building on 64-bit platform) and cross platform (when building on 32-bit platform but targeting a 64-bit platform).

Added the following folders:

```
+ <TargetRTS>/config/NT40T.x64-VisualC++-10.0/
+ <TargetRTS>/config/NT40T.x64-VisualC++-9.0/
+ <TargetRTS>/config/NT40T.x86_x64_cross-VisualC++-10.0/
+ <TargetRTS>/config/NT40T.x86_x64_cross-VisualC++-9.0/
+ <TargetRTS>/config/NT40T.x86-VisualC++-11.0/
+ <TargetRTS>/config/NT40T.x86-VisualC++-10.0/
+ <TargetRTS>/config/NT40T.x86-VisualC++-9.0/
...
```

Updated Copyright texts

Updated a comment holding copyright note in all files.

Fix problem in main.mk to make it work for MinGW

Fixed make rules for $(ACTOR_LIB) and $(DATA_LIB). Replaced "%A.olist" with $(A_OBJECTS) and "%D.olist" with $(D_OBJECTS)

This file was updated:

```
<TargetRTS>/src/main.mk
```

Update stack size settings for Connexis on SLED10x64T

During testing we found a crash in gethostbyname() function call on Linux (SLED10x64T) for models using Connexis. The problem is in the stack size for Connexis threads.

gethostbyname() allocates its return value on the stack and if there is no more space, the function crashes.

Default stack sizes are set up in <TargetRTS>/target/SLED10x64T/RTDcsTarget.h:

```
// Stack size of Connexis Transport thread
#define CNXTTP_STACK 8000
// Stack size of Connexis Helper thread
#define CNXTHTP_STACK 16000
// Recommended minimum stack size for Connexis DCS and Locator Capsules
#define CNXDTP_STACK 8000
// Stack size of Connexis debug Agent thread
#define CNXATP_STACK 8000
// Stack size of Connexis ORB threads
#define CNXODP_STACK 10000
```

We have increased the default stack sizes for SLED 64.

Better error reporting in Perl scripts

These files were updated:

```
<TargetRTS>/codegen/rtcomp.pl
<TargetRTS>/codegen/rterror.pl
```

Added support for Quantify and Pure Coverage in rtlink.pl

These files were updated:

```
<TargetRTS>/codegen/rtlink.pl
```

Improved configuration of paths for GNU compiler in setup.pl

These files were updated:

```
<TargetRTS>/config/NoRTOSS.sparc-gnu-2.8.1/setup.pl
<TargetRTS>/config/NoRTOSS.x86-gnu-3.2/setup.pl
```

Added class-scoped state entry operation rtgStateEntry

Support has been added for a class-scoped state entry operation for capsules and passive classes with state machines.

For capsules, a virtual operation rtgStateEntry has been added to the base class for all capsules, RTActor:

```
class RTActor
{ ... protected: virtual void rtgStateEntry( void ); ... }
```

This operation has a default implementation in the RTActor class:

```
void RTActor::rtgStateEntry( void )
{
// by default class-scoped state entry operation does nothing
}
```

This operation is called in RTActor::enterState(int s) right before the local state-entry operation enterStateV():

```
setCurrentState( s );
rtgStateEntry(); //class-scoped state entry operation
enterStateV();
```

To define a class-scoped state-entry operation for a capsule, you need to add an operation

```
void rtgStateEntry()
```

to it. It’s important to mark it as Virtual so it will be generated as a virtual function (otherwise it will not work as intended). This operation can be inherited or redefined as a normal virtual operation. If a capsule does not define rtgStateEntry, the default implementation from RTActor will be called and it will do nothing.

A typical usage of this feature could be to trace the state name whenever a state is entered.

For passive classes with state machines the support for a class-scoped state entry operation is handled by the code generator. It checks if the passive class contains an operation named rtgStateEntry, local or inherited, and includes the call to it into the generated code.

These files were updated:

```
<TargetRTS>/include/RTActor.h
<TargetRTS>/src/RTActor/enterState.cc
<TargetRTS>/src/RTActor/enterStateV.cc
```

Passing information about initiator to some RTActor functions

This change was made for better error reporting, by setting errors to the correct context.

These files were updated:

```
<TargetRTS>/include/RTActor.h
<TargetRTS>/include/RTActorRef.h
<TargetRTS>/include/RTRelayPort.h
<TargetRTS>/src/RTActor/removeAllImports.cc
<TargetRTS>/src/RTActor/removeImport.cc
<TargetRTS>/src/RTActor/unbind.cc
<TargetRTS>/src/RTActorRef/bind_ports.cc
<TargetRTS>/src/RTActorRef/deport_one.cc
<TargetRTS>/src/RTActorRef/deport_RTActor.cc
<TargetRTS>/src/RTActorRef/destroy_RTActor.cc
<TargetRTS>/src/RTRelayPort/unreserve.cc
```

Added support for unsigned integer basic data types to ascii encoding/decoding and logging functions

These files were updated:

```
<TargetRTS>/include/RTAsciiDecoding.h
<TargetRTS>/include/RTAsciiEncoding.h
<TargetRTS>/include/RTDecoding.h
<TargetRTS>/include/RTEncoding.h
<TargetRTS>/include/RTLog.h
<TargetRTS>/include/RTObject_class.h
<TargetRTS>/src/MANIFEST.cpp
```

These files were added:

```
+ <TargetRTS>/src/RTAsciiDecoding/get_unsignedchar.cc
+ <TargetRTS>/src/RTAsciiDecoding/get_unsignedint.cc
+ <TargetRTS>/src/RTAsciiDecoding/get_unsignedlong.cc
+ <TargetRTS>/src/RTAsciiDecoding/get_unsignedshort.cc
+ <TargetRTS>/src/RTAsciiEncoding/put_unsignedchar.cc
+ <TargetRTS>/src/RTAsciiEncoding/put_unsignedint.cc
+ <TargetRTS>/src/RTAsciiEncoding/put_unsignedlong.cc
+ <TargetRTS>/src/RTAsciiEncoding/put_unsignedshort.cc
+ <TargetRTS>/src/RTLog/show_unsignedshort.cc
+ <TargetRTS>/src/RTLog/show_unsignedlong.cc
+ <TargetRTS>/src/RTLog/show_unsignedint.cc
+ <TargetRTS>/src/RTLog/show_unsignedchar.cc
+ <TargetRTS>/src/RTLog/log_unsignedshort.cc
+ <TargetRTS>/src/RTLog/log_unsignedlong.cc
+ <TargetRTS>/src/RTLog/log_unsignedint.cc
+ <TargetRTS>/src/RTLog/log_unsignedchar.cc
+ <TargetRTS>/src/RTObject_class/unsignedshort.cc
+ <TargetRTS>/src/RTObject_class/unsignedlong.cc
+ <TargetRTS>/src/RTObject_class/unsignedint.cc
+ <TargetRTS>/src/RTObject_class/unsignedchar.cc
```

Added more accurate error message about no free ports triggered by FrameImport

A new, more specific, error message has been added and may be reported during frame import. Now the following message will be printed: “Not enough free ports.” instead of the more general one "Not a compatible subclass.".

These files were updated:

```
<TargetRTS>/include/RTController.h
<TargetRTS>/src/RTActorRef/import.cc
<TargetRTS>/src/RTController/strerror.cc
```

Added support for IPv6 protocol

This file was updated:

```
<TargetRTS>/include/RTTcpSocket.h
```

Print information about port index when receiving unexpected message to port

The port index has been added to the “Unexpected signal” error message.

This file was updated:

```
<TargetRTS>/src/RTActor/badMessage.cc
```

Improved TargetRTS multi-threaded mode (when USE_THREADS is defined)

These files were updated:

```
<TargetRTS>/src/RTController/dt.cc
<TargetRTS>/src/RTPeerController/shutdown.cc
<TargetRTS>/src/RTProtocol/reply.cc
<TargetRTS>/src/RTProtocol/sendOne.cc
```

Added support for 64-bit MinGW GCC 8.1.0

The TargetRTS now contains new config and libset files and prebuilt libraries for supporting 64-bit compilation with gcc 8.1.0 from MinGW.

Added the following folders:

```
+ <TargetRTS>/config/MinGwT.x64-MinGw-gnu-8.1.0/
+ <TargetRTS>/libset/x64-MinGw-gnu-8.1.0/
+ <TargetRTS>/lib/MinGwT.x64-MinGw-gnu-8.1.0/
```

Added support for Visual Studio 2017 (15.0)

The TargetRTS now contains new config and libset files and prebuilt libraries for supporting compilation with Visual Studio 2017 (ver 15.0), for both 32 (x86) and 64 bit (x64).

Added the following folders:

```
+ <TargetRTS>/config/WinT.x64-VisualC++-15.0/
+ <TargetRTS>/config/WinT.x86-VisualC++-15.0/
+ <TargetRTS>/libset/x64-VisualC++-15.0/
+ <TargetRTS>/libset/x86-VisualC++-15.0/
+ <TargetRTS>/lib/WinT.x64-VisualC++-15.0/
+ <TargetRTS>/lib/WinT.x86-VisualC++-15.0/
```