Orocos Toolchain 2.6 Release Series Changes, New Features, and Fixes

Open RObot COntrol Software

Table of Contents

This document provides a quick overview of what changed between the Orocos Toolchain 2.5 and version 2.6.

1. Upgrading from RTT 1.x to Toolchain 2.x

Please consult the online wiki for all available information about the 1.x to 2.x transition: The Road to RTT 2.0 [http://www.orocos.org/wiki/rtt/rtt-20]. The Real-Time Toolkit is no longer released as a single entity, but as part of a complete set of tools and libraries for creating real-time applications. See also the changes documents of the previous major releases:

- Toolchain 2.0 Changes [http://www.orocos.org/stable/documentation/rtt/v2.0.x/doc-xml/orocos-rtt-changes.html]

- Toolchain 2.1 Changes [http://www.orocos.org/stable/documentation/rtt/v2.1.x/doc-xml/orocos-rtt-changes.html]

- Toolchain 2.2 Changes [http://www.orocos.org/stable/documentation/rtt/v2.2.x/doc-xml/orocos-rtt-changes.html]

- Toolchain 2.3 Changes [http://www.orocos.org/stable/documentation/rtt/v2.3.x/doc-xml/orocos-rtt-changes.html]

- Toolchain 2.4 Changes [http://www.orocos.org/stable/documentation/rtt/v2.4.x/doc-xml/orocos-rtt-changes.html]

- Toolchain 2.5 Changes [http://www.orocos.org/stable/documentation/rtt/v2.5.x/doc-xml/orocos-rtt-changes.html]

2. Important Caveats

- This release is binary incompatible with RTT 2.5.x. You need to recompile the whole toolchain and all your applications in order to use this release.

  Note

  Take care that all previously built components and plugins in the <install>/lib/orocos directory and subdirectories are removed, since the component and plugin loaders may get confused if older versions are found.

- The rttlua DeploymentComponent's name has been changed from "deployer" to "Deployer" in order to be compatible with the existing deployer application. This change only influences Lua users and requires each occurence of getPeer("deployer") to be changed by getPeer("Deployer").

  You can do the renaming in your existing source trees with this Shell command: **find . ! -path "*.git*" -name "*.lua" -print0 | xargs -0 sed -i "s/\"deployer\"/\"Deployer\"/ g;s/\'deployer\'/\'Deployer\'/g"**

- There was an API change in the (RTT internal) types::TypeInfo class. If you inherited directly or indirectly from this class, you'll probably need to inherit from types::TypeInfoGenerator instead and fabricate a installTypeInfoObject() function. There is generally no change needed on your side if you inherited from types::TemplateTypeInfo, types::StructTypeInfo or similar classes, since these classes have implemented installTypeInfoObject() function for you and inherit from types::TypeInfoGenerator.

- There was an API change in the (RTT internal) marsh::PropertyLoader class. It requires now a TaskContext or Service in the constructor and no longer as argument to the rest of the API.

- *We no longer load component libraries with the RTLD_GLOBAL flag.*

  This means you must link with all required libraries at link time, and not rely on symbol resolution during the loading by the deployer.

3. ROS Users

- The Orocos Toolchain 2.6.x will only be released in ROS Fuerte and later.

- We fixed issues of conflicting typekits (see more on that below). This means that its allowed now to load typekits from ROS and non-ROS packages, describing the same structs.

4. Improvements

4.1. Building

- *ROS_ROOT is no longer influencing your build, unless you're in a ROS_PACKAGE_PATH* The last cornercases of ROS_ROOT influencing our build behavior have been ironed out in this release.

This fixes integration issues with AutoProj.

- *RTT and OCL builds with clang++ 3.0*

This allows you to speed up RTT builds significantly. When you enabled CORBA, only the OMNIORB transport compiles fine in clang.

- *It's now possible to create applications that setup a CORBA proxy, without linking to TAO/ OMNIORB directly, but only to the RTT corba transport.* See the corba::TaskContextFactory class.

4.2. TaskContext and Services

- *A new connection type has been introduced, CIRCULAR_BUFFER*

This new type can be used in the ConnPolicy.type field in order to have circular buffers instead of the classical non-circular buffer.

- *Exceptions thrown in operations are now detected and handled.*

When an operation threw an exception, this resulted in undefined behavior. The exception is now caught by RTT and the caller side receives an std::runtime_exception exception. Passing exceptions is not supported, so this is a last-resort solution to cope with misbehaving operations.

- *The ComponentLoader is from now on part of RTT instead of OCL.*

This means you no longer need a DeploymentComponent to import typekits or component packages.

4.3. Data Types, Toolkits and Typekits

- *We restructured typekits such that it is now allowed to load multiple typekits for the same type.*

If these typekits name the type differently, the names will function as an alias. Also, TypeInfo object pointers are now constant and can no longer become invalid in a running application. The restructuring allows one to install a TypeInfo object and later-on add features such as type decomposition or transports. The classical TypeInfo subclasses all have been adapted, but if you created your own TypeInfo variant, you will need to update your code minimally.

See the Caveats section above in case you wrote a custom typekit.

- *The internal function getMember is now much more efficient and faster.*

There's now also a real-time version of getMember, if you provide it a ReferenceDataSource such that a reference to the member can be given without allocating a DataSource for it.

4.4. Operating Systems

- Creating a Linux thread now prints the TID or PID instead of the POSIX id, such that RTT threads can be correlated to normal Unix tools such as top or ps.

5. Orocos Component Library (OCL)

5.1. Lua scripting

- rttlua and tlsf-rttlua now support CORBA and Log4cpp logging as well.

- tlsf-rttlua is now capable of executing RTT operations in hard real-time, as long as they are assigned to an operation variable in a non-critical-time section.

- Consult the LuaCookbook [http://www.orocos.org/wiki/orocos/toolchain/orocostool-chainluacookbook] for the latest changes

5.2. Reporting

- The reporting component has been rewritten (again) boosting its performance sometimes a 1000-fold (but at least a 100-fold), especially when using the NetCDF reporting component. We can now reliably log a dozen ports at 1kHz, using only 10% cpu on an Intel i7 system.

5.3. Deployment

- A *-d or --daemon* flag has been added to the deployer to let it run in the background. You can use the new signal catching functions in the deployer API to catch shutdown/restart signals.

- Two new operations were added: setFileDescriptorActivity and connectOperations.

1. About Orocos

Please send your Orocos questions to orocos-users at lists.mech.kuleuven.be [mailto: orocos-users at lists.mech.kuleuven.be].

These pages are maintained by the Orocos team [http://www.orocos.org/orocos/whatis].

For questions related to the use of the Orocos Software, please consult these web pages and the Orocos RTT manuals [http://www.orocos.org/rtt]. If that fails, the orocos-dev at lists.mech.kuleuven.be [mailto:orocos-dev at lists.mech.kuleuven.be] mailing list might help. Please send comments on these web pages and the development of Orocos to our developer mailing list at orocos-dev at lists.mech.kuleuven.be [mailto:orocos-dev at lists.mech.kuleuven.be]. All of our lists have public archives [http://lists.mech.kuleuven.be/mailman/listinfo/orocos-users] (dev public archive [http://lists.mech.kuleuven.be/mailman/listinfo/orocos-dev]) .

Copyright (C) The SourceWorks

Verbatim copying and distribution of this entire article is permitted in any medium, provided this notice is preserved.