RTA-OS3.0

Getting Started Guide

Copyright

Document: 10384-GS-1.0.0

Contents

1 Introduction

Welcome to RTA-OS3.0!

In this guide you will find out how to install RTA-OS3.0, check that your installation works and learn about what RTA-OS3.0 can do.

RTA-OS3.0 is a small and fast real-time operating system that conforms to both the AUTOSAR OS R3.0 and OSEK/VDX OS 2.2.3 standards. The operating system is configured and built on a PC for use on a target hardware platform.

There are two major parts to RTA-OS3.0:

- PC Tools

  The RTA-OS3.0 tools are target independent and include:

  rtaoscfg a graphical configuration editor for configuring RTA-OS3.0.

  rtaosgen a command line tool for generating an RTA-OS3.0 kernel library based on your configuration. This includes the parts of the RTA-OS3.0 kernel that are shared across all targets (approximately 95% of total kernel code).

- Port Plug-ins

  RTA-OS3.0 port plug-ins are used to customize RTA-OS3.0 for different target microcontroller and compiler combinations. The PC tools support multiple port plug-ins and you can switch between ports easily at configuration time.

  Your RTA-OS3.0 tools installation includes a port plug-in for a “Virtual Target” called VRTA. VRTA allows you to build RTA-OS3.0 applications that run on your PC without the need for embedded target hardware, using an ANSI C compiler.

1.1 About You

You are a trained embedded systems developer who wants to build real-time applications using a pre-emptive operating system. You should have knowledge of the C programming language, including the compilation, assembling and linking of C code for embedded applications with your chosen tool chain. Elementary knowledge about your target microcontroller, such as the start address, memory layout, location of peripherals as so on, is essential.

You should also be familiar with common use of the Microsoft Windows®2000, Windows®XP or Windows®Vista operating systems, including installing software, selecting menu items, clicking buttons, navigating files and directories.

You should read this guide before you install RTA-OS3.0 and before you read any other manuals.

1.2 Document Conventions

The following conventions are used in this guide:

Choose **File > Open**.	Menu options are printed in **bold, blue** characters.
Click **OK**.	Button labels are printed in **bold** characters
Press <Enter>.	Key commands are enclosed in angle brackets.
The “Open file” dialog box appears	The names of program windows, dialog boxes, fields, etc. are enclosed in double quotes.
`Activate(Task1)`	Program code, header file names, C type names, C functions and RTA-OS3.0. Component API call names all appear in the `courier` typeface.
See Section 1.2.	Hyperlinks through the document are shown in red letters.
	Functionality that is provided in RTA-OS3.0 but it may not be portable to another AUTOSAR OS implementation is marked with the ETAS logo.
	Caution! Notes like this contain important instructions that you must follow carefully in order for things to work correctly.

1.3 References

OSEK is a European automotive industry standards effort to produce open systems interfaces for vehicle electronics. For details of the OSEK standards, please refer to:

`http://www.osek-vdx.org`

AUTOSAR (AUTomotive Open System ARchitecture) is an open and standardized automotive software architecture, jointly developed by automobile manufacturers, suppliers and tool developers. For details of the AUTOSAR standards, please refer to:

`http://www.autosar.org`

2 Installing RTA-OS3.0

2.1 Preparing to Install

Before installing RTA-OS3.0 check that all the items have been delivered. You should have been supplied with an RTA-OS3.0 tools installation and, if you plan to run RTA-OS3.0 on embedded target hardware, one (or more) port installations.

RTA-OS3.0 is provided as a downloadable electronic installation image which you obtain from the ETAS Web Portal. You will have been provided with access to the download when you bought the product. You may optionally have requested an installation CD which will have been shipped to you. In either case, the electronic image and the installation CD contain identical content.

The RTA-OS3.0 tools are supplied as a single installer. Each different port is supplied as a separate installer. A working version of RTA-OS3.0 requires the installation of the tools and at least one port.

2.1.1 Hardware Requirements

You should make sure that you are using at least the following hardware before installing and using RTA-OS3.0 on a host PC:

- 1GHz Pentium (or higher) IBM compatible PC.
- 512Mb RAM.
- 500Mb hard disk space.
- CD-ROM or DVD drive (Optional)
- Ethernet card.

2.1.2 Software Requirements

RTA-OS3.0 requires that your host PC has one of the following versions of Microsoft Windows installed:

- Windows 2000 (Service Pack 3 or later)
- Windows XP (Service Pack 2 or later)
- Windows Vista

RTA-OS3.0 does not require any special user privileges for installation.

The tools provided with RTA-OS3.0 require Microsoft's .NET Framework v2.0. You should ensure that this has been installed before installing RTA-OS3.0. The .NET framework is not supplied with RTA-OS3.0 but is freely available from http://www.microsoft.com/net/Download.aspx.

You will need to obtain a license key to run RTA-OS3.0 after installation. Licensing is managed by the ETAS Licence Manager. For further details see Section 2.5.

2.2 Installation

The RTA-OS3.0 installer contains the offline tools and a port plug-in for VRTA, a "virtual target" for RTA-OS3.0 that enables the creation of Windows-hosted applications that emulate the behavior of applications running on microcontroller hardware. VRTA is a complete implementation of RTA-OS3.0 for a PC, so you can develop, test and debug RTA-OS3.0 configurations without needing access to embedded tools of target hardware. VRTA is pre-configured to work with several popular PC-hosted C/C++ compilers.

If purchased, you will also have been supplied with additional installers for embedded port plug-ins.

2.2.1 Tools

You must install the RTA-OS3.0 PC tools first before installing VRTA or any port plug-ins - this will create the necessary directory structures.

1. Either
   - Double click the executable image; or
   - Insert the RTA-OS3.0 CD into your CD-ROM or DVD drive.

     If the installation program does not run automatically then you will need to start the installation manually, by navigating to the root directory of your CD/DVD drive and double-clicking on `autostart.exe` to start the setup.
2. Follow the on-screen instructions to install RTA-OS3.0.

By default, RTA-OS3.0 installs into `C:\ETAS\RTA-OS3.0`. During the install, you will be given the option to change the folder to which RTA-OS3.0 is installed. You can choose any folder on the system but the choice of anything other than the default may require that the example application be changed before it can be built.

2.2.2 VRTA

A default installation of RTA-OS3.0 installs a port plug-in for VRTA into `<install dir>\Targets\VRTA_n.n.n`. VRTA also installs a set of runtime libraries and executables that provide run-time support for VRTA-based applications into `<install dir>\Bin`.

The VRTA port plug-in and associated tools are optional - you can choose not to install VRTA during the installation process.

You must install at least one port plug-in to be able to use the RTA-OS3.0 development tools. It is recommended that you accept the default installation of the VRTA port plug-in.

The VRTA run-time libraries and executables must be on your path for VRTA applications to run. It is recommended that you add `<install dir>\Bin` to your Windows PATH environment variable. If you have installed RTA-OS3.0 on a PC that also includes an installation of RTA-OSEK then you must ensure that the path to RTA-OSEK is placed before the path to RTA-OSEK.

Compilers for VRTA

VRTA is configured to work with several popular PC C/C++ compilers, including, but not limited to:

- MinGW / gcc
- Microsoft Visual Studio 2005
- Microsoft Visual Studio 2008

A full list of supported compilers is provided in the *RTA-OS3.0 VRTA Port Guide*.

These compilers are not supplied as part of the RTA-OS3.0 installation.

The MinGW (Minimalist GNU for Windows) C compiler is available from `http://www.mingw.org`. This is the default compiler used for building VRTA-based applications.

Microsoft provides an 'Express' Edition of the Visual Studio C++ compiler which is freely downloadable from `http://www.microsoft.com/express/vc`.

2.2.3 Embedded Port Plug-ins

Embedded port plug-ins are installed in the same way as the tools:

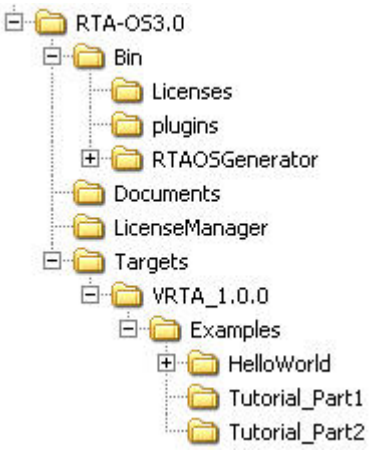


Figure 2.1: Installation Directory Structure

1. Either

   - Double click the executable image; or
   - Insert the RTA-OS3.0 CD into your CD-ROM or DVD drive.

     If the installation program does not run automatically then you will need to start the installation manually. Navigate to the root directory of your CD/DVD drive and double-clicking on `autostart.exe` to start the setup.

2. Follow the on-screen instructions to install the port plug-in.

Ports are installed by default into `C:\ETAS\RTA-OS3.0\Targets`. During the install, you will be given the option to change the folder to which RTA-OS3.0 ports installed, however, ports must be installed `<install dir>\Targets` where `<install dir>` is the installation directory for the RTA-OS3.0 tools.

2.3 What is Installed?

After installing RTA-OS3.0, a number of new folders will be created. The location of these folders depends on where you decided to install the files. The default location is:

`C:\ETAS\RTA-OS3.0`

Figure 2.1 shows the directory structure following an installation.

Filename	Contents
\Bin	Executable programs.
\Bin\Licenses	License signature files that tell the ETAS Licence Manager which licenses are required for the installed programs. The license themselves are installed elsewhere, see Section 2.5.
\Bin\plug-ins	GUI plug-ins for the **rtaoscfg** configuration tool.
\Documents	User documentation.
\License Manager	The installer for the ETAS License Manager.
\Targets	Port plug-ins - one per sub-directory. All port-specific parts, including the *RTA-OS3.0 Target/Compiler Port Guide* for the port, are included in the sub-directory.
\Targets\<port>\Examples	Example applications, one per sub-directory.

Each port plug-in and its associated documentation is installed in its own sub-directory under Targets. These directories have the following naming convention:

<Target><CompilerVendor>_n.n.n[1]

where n.n.n indicates the version of the port plug-in. You can, if required, have multiple different versions of the same target installed at the same time. Each port directory includes the port plug-in DLL.

All user documentation is distributed in PDF format which can be read using Adobe Acrobat Reader. Adobe Acrobat Reader is not supplied with RTA-OS3.0 but is freely available from http://www.adobe.com.

2.4 Running RTA-OS3.0 from a Network Drive

The RTA-OS3.0 tools are Windows .NET applications. By default a .NET assembly (application) *cannot* be run from a network drive due to limited permissions in the Windows security policy so this means that the RTA-OS3.0 tools will not run from network drive without additional configuration.

You can modify the Windows security policy to allow RTA-OS3.0 to run from a network in two ways using either:

- the Windows Control Panel; or

[1] Note that the VRTA port plug-in is an exception to this naming convention as it can support multiple compilers.

- the Microsoft command-line tool **CasPol.exe** which is distributed with the Microsoft .NET framework Control Panel

The Control Panel method is simple, but offers less control than that provided by the command-line method. You should consult your IT support department for recommendations on what levels of security are allowed at your workplace.

2.4.1 Using the Control Panel

Navigate to **Control Panel → Administrative Tools →.NET Framework Configuration x.x → Runtime Security Policy → Adjust Zone Security** and adjust the setting for "Local Intranet" to "Full".

2.4.2 Using the Command-line

The **CasPol.exe** program is used to change the security policy for the share from which the application will be run. It can be found in `C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\CasPol.exe`. You need to perform the following steps:

1. Disable policy change prompt using **CasPol.exe -pp off**. The `-pp off` option disables the policy change prompt.
2. Change policy for share using **CasPol.exe -m -ag 1.2 -url "file:<share_name>/*" FullTrust** where:

Argument	Description
-m	Operate on machine level
-ag 1.2	Add a code group under group 1.2. In the default policy, Group 1.2 is the LocalIntranet group, so the new code group that we're creating will only be checked if the file comes from the intranet.
-url:...	Match any file within the specified URL, where <share_name> should be replaced with the name of the share, e.g. `\networkdrive\ETAS\RTA-OS3.0`.
FullTrust	The permission set to grant assemblies that match the code group. In this case, FullTrust.

3. Re-enable policy change prompt using **CasPol.exe -pp on**. The `-pp off` option disables the policy change prompt.

2.5 Licensing

RTA-OS3.0 is protected by FlexNet license management software. RTA-OS3.0 Licenses are installed and managed by the ETAS License Manager. This is a

common ETAS software component that is installed in parallel with RTA-OS3.0 installation. The ETAS license manager stores data in the following locations:

- The ETAS License Manager and the FlexNet support software:

  ```
  C:\Program Files\Common Files\ETAS\Licensing
  ```

- The license files:

  ```
  C:\Documents and Settings\All Users\Application Data\ETAS\
      FlexNet
  ```

If you do not have a license key then please contact your local ETAS sales representative. Contact details can be found in Chapter 5.

2.5.1 License Types

ETAS can provide three types of licenses:

Machine-named licenses limit the use of RTA-OS3.0 to the PC on which it is installed using the Host ID of the your host PC. The Host ID is the MAC address of your PC's Ethernet controller.

To obtain a machine-named licence you need to provide ETAS with the following information:

1. The Host ID (MAC address) of the host PC.

The ETAS License Manager provides support to get this information - see Section 2.5.2 for more details.

User-named licenses limit the use of RTA-OS3.0 to the user for which it is installed.

To obtain a user-named licence you need to provide ETAS the following information:

1. Your Windows user-name for your network domain.

The ETAS License Manager provides support to get this information - see Section 2.5.2 for more details.

Concurrent licenses allow a specified number of (named) users across multiple sites to use RTA-OS3.0 simultaneously. They are sometimes called "floating" licenses because the licence can "float" between users.

Concurrent licenses are allocated to PCs by a license server. The license server and associated infrastructure will be provided and managed by your IT department. ETAS provides the license that can be served.

Figure 2.2: Missing License Warning

To obtain a concurrent licence you need to provide ETAS the following information:

1. The name of your FlexNet licence server
2. The TCP/IP port over which license are served by your FlexNet licence server. A default installation of the FlexNet licence server uses port 27000.
3. The Host ID (MAC address) of the FlexNet server.

2.5.2 Using the ETAS License Manager

If you try to use RTA-OS3.0 without a valid license then the ETAS Licence Manager will display the warning in Figure 2.2.

Clicking “Continue” starts the ETAS License Manager application. If you disable this warning then the ETAS License Manager will start automatically each time you try to use an unlicensed feature of any ETAS product.

Until a valid license is installed the ETAS License Manager will display the RTA-OS3.0 license state as `NOT AVAILABLE` and you will not be able to use any of the tools as shown in Figure 2.3.

Licence Key Installation

All licenses are supplied in an ASCII text file, which will be sent to you on completion of a valid license agreement.

License files are installed onto your host PC from the **File → Add License File** menu. When a valid license has been installed, ETAS License Manager will display the license version, status, expiration date and source as shown in Figure 2.4.

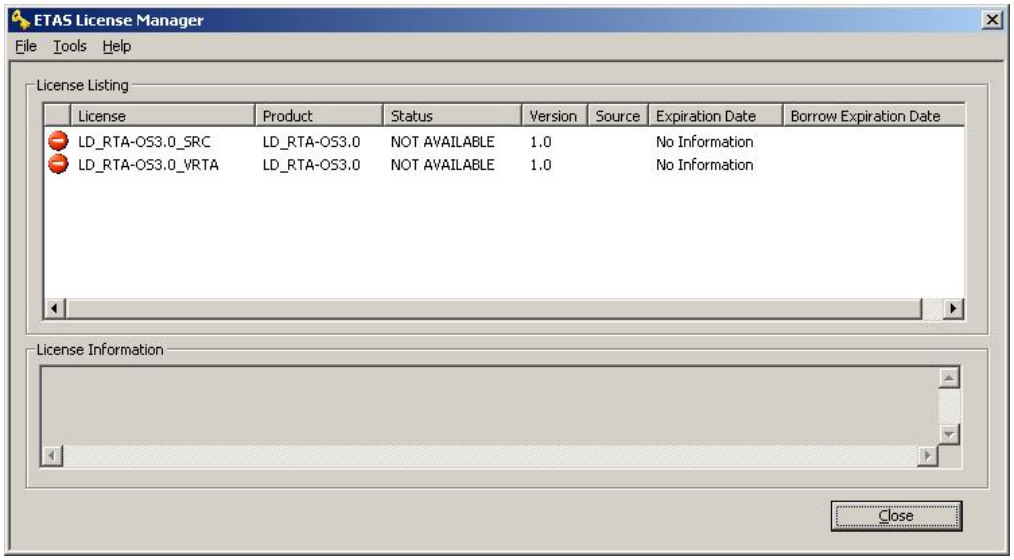


Figure 2.3: Unlicensed RTA-OS3.0 Installation

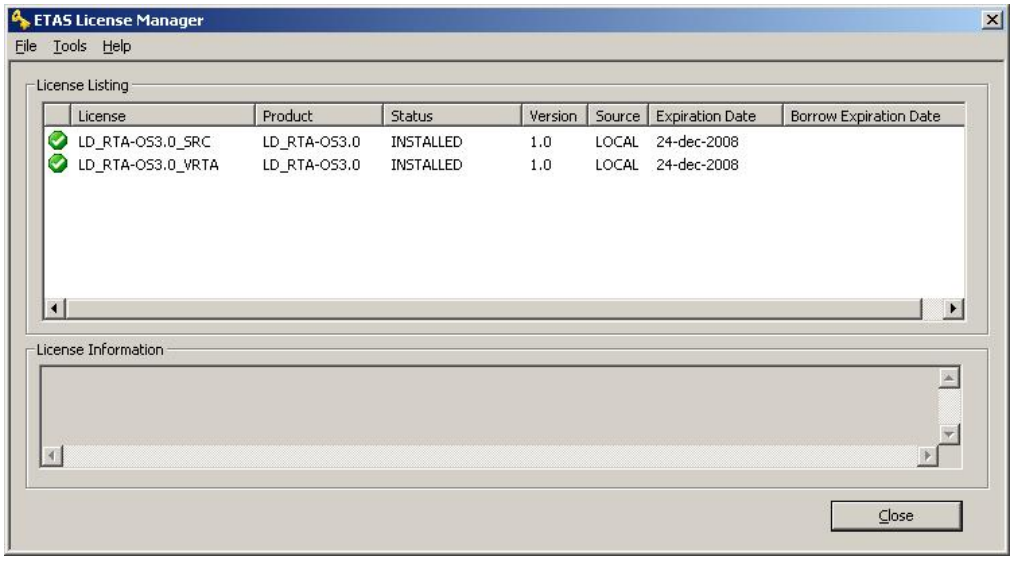


Figure 2.4: Licensed features for RTA-OS3.0

Figure 2.5: Obtaining Licence Information

Obtaining Machine and User-Named Licence Information

If you need to request a machine-named or user-named licence from ETAS then you can use the ETAS License Manager to get the necessary information you will need to provide to ETAS and (optionally) store this in a text file.

This can be performed using the **Tools → Obtain License Info** menu. For machine-named licenses you can then select the network adaptor which provides the Host ID (MAC add dress) that you want to use as shown in Figure 2.5. For user-based license, the ETAS Licence Manager automatically identifies the Windows username for the current user.

Selecting “Get License Info” tells you the Host ID and User information and lets you save this as a text file to a location of your choice.

2.5.3 Troubleshooting Licenses

rtaosgen and **rtasoscfg** will report an error if you try to use a feature for which a correct license key cannot be found. If you think that you should have a license for a feature but RTA-OS3.0 appears not to work then should follow the trouble shooting steps below before contacting ETAS:

Can RTA-OS3.0 see the license?

The ETAS license manager looks for the license file in `C:\Documents and Settings\All Users\Application Data\ETAS\FlexNet`.

If you have installed other FlexNet licensed products then you may have the environment variable `LM_LICENCE_FILE` set. `LM_LICENCE_FILE` tells FlexNet

where to look for licenses. If this variable is set then it overrides the default search path.

To solve this problem you will need to add the path to your RTA-OS3.0 license file to the LM_LICENCE_FILE environment variable. You can do this using Windows Control Panel. Open **System → Advanced → Environment Variables** and look in the User and/or System Variables for LM_LICENCE_FILE. Append the path to your RTA-OS3.0 license to the variable.

Is the license valid?

You may have been provided with a time-limited license (for example, for evaluation purposes) and the license may have expired. You can check that the expiry date for your licensed features has not elapsed using the ETAS License Manager.

If your license has elapsed then please contact your local ETAS sales representative to discuss your options.

Does the Ethernet MAC address match the one specified?

If you have a machine based licence then it is locked to a specific MAC address. You can find out the MAC address of your PC by running the Microsoft program **ipconfig /all** at a Windows Command Prompt. The *Physical Address* reported should be identical to the MAC address in your license file. Section 2.5.2 shows you how to get the MAC address of your host PC.

You can check that the MAC address in your license file by opening your license file in a text editor and looking for the HOSTID. The physical address reported by ipconfig /all and the HOSTID in the license file should be identical. If these are not identical then you do not have a valid license for your PC. You should contract your local ETAS sales representative to discuss your options.

Is your Ethernet Controller enabled?

If you use a laptop and RTA-OS3.0 stops working when you disconnect from the network then you should check your hardware settings to ensure that your Ethernet controller is not turned off to save power when a network connection is not present. You can do this using Windows Control Panel. Select **System → Hardware → Device Manager** then select your Network Adapter. Right click to open **Properties** and check that the Ethernet controller is not configured for power saving in **Advanced** and/or **Power management** settings.

Still not fixed?

If you have not resolved your issues, after confirming these points above, please contact ETAS technical support. The contact address is provided in

Section 5.1. You must provide the contents and location of your license file and your Ethernet MAC address.

2.6 Verifying your Installation

Now that you have installed RTA-OS3.0 and have obtained and installed a valid license key you can check that things are working. You can verify that the PC tools installation has worked by running the RTA-OS3.0 code generator from the command line:

```
C:\> rtaosgen --targets:?
```

If the installation has been successful then the tool will run and tell you whether it can see the license file or not and provide a list of all installed target plug-ins.

To verify that a port plug-in is working correctly, each target includes a “Hello World” application that you can use to check that an RTA-OS3.0 library can be built for you target. For specific instructions you need to read the *RTA-OS3.0 Target/Compiler Port Guide* for your particular target. For example, if you have installed VRTA port plug-in suggested in Section 2.2.2 then you need to read the *RTA-OS3.0 VRTA Port Guide*.

3 Developing Applications with RTA-OS3.0

The process for using RTA-OS3.0 in your application involves three steps:

1. Configure the features of the OS you want to use
2. Generate a customized RTA-OS3.0 kernel library
3. Use the OS in your application

3.1 Configuration

RTA-OS3.0 is statically configured. This means you need to declare every task and interrupt that you need over the entire operation of your application at configuration time, together with any critical sections, synchronization points, counters etc.

All configurations are held in XML files that conform to the AUTOSAR standard. This means that you are free to use other 3rd party tools to provide the XML configuration files for the OS. If you look at one of these files in the examples provided by the target plug-ins then you'll realize that the configuration language isn't that easy to read.

To help you with configuration, RTA-OS3.0 includes **rtaoscfg**, a graphical configuration editor for configuring your RTA-OS3.0 application. **rtaoscfg** accepts any AUTOSAR XML file as input and allows you to edit the OS-specific parts of configuration. If the input file contains both OS and non-OS specific configuration then only the OS configuration will be modified. Figure 3.1 shows the **rtaoscfg** configuration tool. Further details about how to use **rtaoscfg** are given in the *RTA-OS3.0 User Guide*.

3.2 Library Generation

Before you can use RTA-OS3.0 in your application you need to generate an RTA-OS3.0 kernel library and associated header file for your configuration. Library generation is done using a command-line tool called **rtaosgen**. **rtaosgen** generates a customized RTA-OS3.0 kernel library for your OS configuration by:

- Analyzing your XML configuration and automatically optimizing the RTA-OS3.0 kernel so that it contains only those features that you will use. This makes RTA-OS3.0 as small and efficient as possible.
- Customizing the optimized kernel to your chosen target using information provided by the target plug-in.

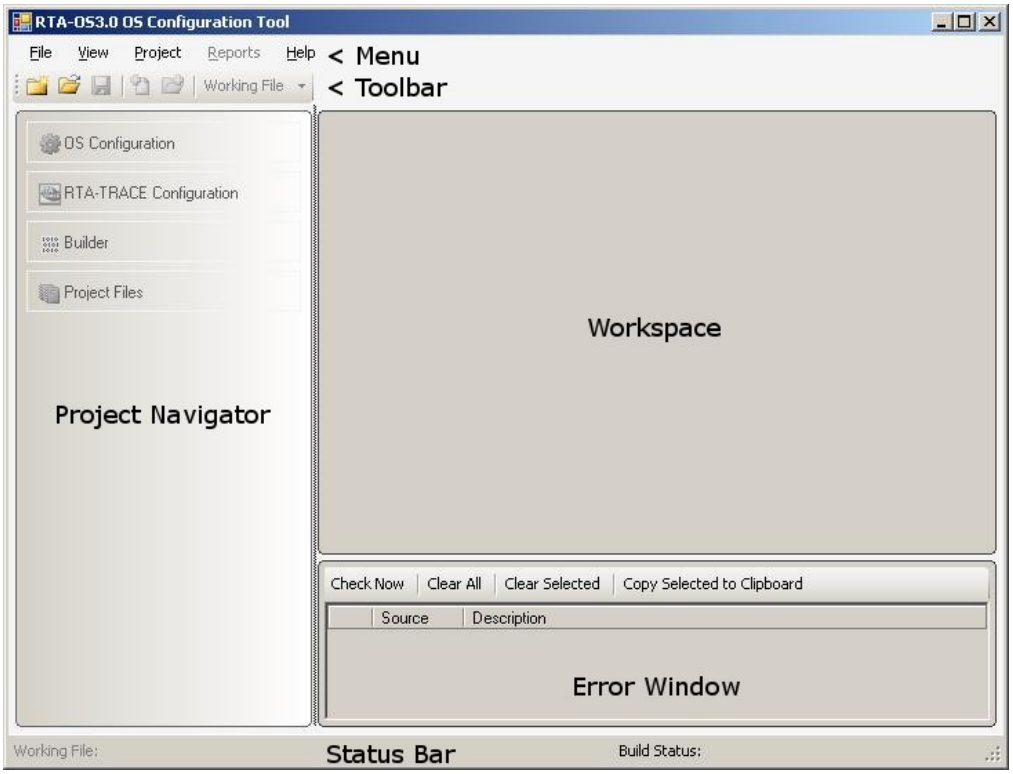


Figure 3.1: The rtaoscfg configuration tool

- Building a kernel library using the same 3rd party tool chain that you are using for your application. This guarantees tool chain compatibility between RTA-OS3.0 and your own code.

3.2.1 Preparing the Tool Chain

To generate a kernel library, **rtaosgen** needs access to the compiler tool chain for your target. **rtaosgen** knows how to run the compiler, assembler, linker and librarian for your target and what options to use. You need only worry about two things:

1. Your compiler tool chain must be accessible on your PATH.
2. Your compiler tool chain must be compatible with RTA-OS3.0.

You can find out if your compiler is on your PATH by opening a Windows Command Prompt and typing `C:\> set` at the command prompt. This will list all of your environment variables. You should see your compiler's executable directory on the PATH variable. If you don't, then you can add your compiler to the path by typing:

```
C:\> set PATH=PATH;<Path to your compiler executable>
```

To check whether you are using a compatible version of the compiler tool chain you should consult the *RTA-OS3.0 Target/Compiler Port Guide* for your port which will tell you which version (or versions) are compatible.

3.2.2 Running **rtaosgen**

rtaosgen is a command line tool. You can invoke it from the Windows command prompt, from a batch file, a make script, Ant script, etc., in fact from anywhere where you can call a Windows executable. The **rtaoscfg** tool can invoke **rtaosgen** from the builder if you prefer using a graphical environment.

3.2.3 Building the library

RTA-OS3.0 is an AUTOSAR basic Software Module and therefore requires access to the standard AUTOSAR header files. You must include the path to the location of the AUTOSAR standard header files when invoking **rtaosgen**. For example, to build a library for the “Hello World” example application for an RTA-OS3.0 target you can type:

```
C:\>rtaosgen --include:PathToAutosarHeaderFiles HelloWorld.xml}
```

If you do not have access to AUTOSAR include files (for example, if you are using RTA-OS3.0 for application development outside of a full AUTOSAR system) then **rtaosgen** can generate them automatically for you. You have to tell RTA-OS3.0 where the generated headers can be found, this is typically in a directory named `Samples\Includes`. The command line in this case would be:

```
C:\>rtaosgen --samples:[Includes] --include:Samples\Includes
   HelloWorld.xml
```

Note that **rtaosgen** does not force you to use a specific extension - you can use any extension you like - however, AUTOSAR recommends that files use `.xml` (short for AutosaR XML). RTA-OS3.0 user documentation adopts this convention.

rtaosgen generates four classes of messages during execution:

Information. These messages tell you useful things about the configuration, for example how many tasks you have configured. **rtaosgen** will generate output files.

Warning. These messages warn you that your configuration will result in an OS that might not behave as you expect. **rtaosgen** will generate output files.

Errors. These messages tell you that there is something wrong with your configuration. **rtaosgen** will stop processing your configuration at a convenient point and no output files will be generated.

Fatal. You will get at most one fatal message. It tells you that there is something fundamentally wrong with either your configuration or **rtaosgen**. **rtaosgen** stops immediately.

3.2.4 Generated Files

When **rtaosgen** runs and terminates without generating any errors or fatal messages then it will have generated the following files:

Filename	Contents
`Os.h`	The main include file for the OS.
`Os_Cfg.h`	Declarations of the objects you have configured. This is included by `Os.h`.
`Os_MemMap.h`	AUTOSAR memory mapping configuration used by RTA-OS3.0 to merge with the system-wide `MemMap.h` file.
`RTAOS.<lib>`	The RTA-OS3.0 library for your application. The extension `<lib>` depends on your target.
`RTAOS.<lib>.sig`	A signature file for the library for your application. The extension `<lib>` depends on your target.

There may be other files generated depending on your chosen port. Consult the relevant *RTA-OS3.0 Target/Compiler Port Guide* for further information.

3.3 Integration

3.3.1 Accessing the OS in your Source Code

To access RTA-OS3.0 in your source code you simply include `Os.h` in every C compilation unit (i.e. every C source code file) where you need to access the OS. The header file is protected against multiple-inclusion.

You can split your source code across files however you like[1]. It is possible to have more than one task and interrupt per source file. However, for the purposes of version control we would recommend that you place each task and interrupt implementation into its own C source file.

[1] Within the limits of what is permitted by the C language.

3.3.2 Implementing Tasks and ISRs

Tasks

For each task that you declared at configuration time you must provide an implementation of the task. Each task needs to be marked using the `TASK(x)` macro. Tasks typically have the following structure:

```
#include <Os.h>
TASK(MyTask){
  /* Do something */
  TerminateTask();
}
```

Category 2 ISRs

Each Category 2 ISR that you declared also needs to be implemented. This is also marked, this time using the `ISR(x)` macro:

```
#include <Os.h>
ISR(MyISR){
  /* Do something */
}
```

A Category 2 ISR handler does not need to execute a return from interrupt call - RTA-OS3.0 does this automatically. Depending on the behavior of interrupt sources on your target hardware you may need to clear the interrupt pending flag as part of your ISR. Please consult the hardware documentation provided by your silicon vendor for further details.

Category 1 ISRs

Each Category 1 ISR that you declared also needs to be implemented. Your compiler will use a special convention for marking a C function as an interrupt. RTA-OS3.0 provides a macro that expands to the correct directive for your compiler. Your Category 1 handler will therefore look something this:

```
CAT1_ISR(MyCat1ISR) {
  /* Do something */
}
```

Starting the OS

RTA-OS3.0 does not take control of your hardware after it comes out of reset so you are free to do any initialization you require without interference from the OS - just as if you were developing an embedded application that doesn't use an operating system. Once you have done hardware initialization then you need to start the OS manually using the `StartOS()` API call, usually in

your `main()` program. RTA-OS3.0 provides the `OS_MAIN()` macro that can be used to indiccate the `main()` program.

```
#include <Os.h>
OS_MAIN(){
  /* Initialize target hardware */
  /* Do any mode management, pre-OS functions etc. */
  StartOS();
  /* Call does not return so you never reach here */
}
```

3.3.3 Interacting with the RTA-OS3.0

You interact with RTA-OS3.0 my making kernel API calls. You can find a complete list of calls in the *RTA-OS3.0 Reference Guide*.

3.3.4 Compiling and Linking

When you compile your code you must make sure that `Os.h` and `Os_Cfg.h` are reachable on your compiler include path. When you link your application you must have `RTAOS.<lib>` on your linker library path.

You will must also ensure that any tool options you use are compatible with those expected by the RTA-OS3.0 library. A full list of mandatory and prohibited toolchain options is provided in the *RTA-OS3.0 Target/Compiler Port Guide* for your port.

3.3.5 Common Problems

If the compiler generates errors related to RTA-OS3.0 then you should check the following:

- `Os.h` and `Os_Cfg.h` are on the compiler include path
- You have rebuilt the kernel library after making any changes
- `RTAOS.<lib>` is on your linker library path

3.3.6 Downloading to your Target

The output of the linker is typically a binary file in some well-known format (e.g. a.out, COFF, ELF or IEEE695). These can typically be read by debuggers, in-circuit emulators or in-circuit programming equipment, although in some cases it is necessary to convert the output from this binary format into a text-based form (such as S-Records or Intel Hex) that can be transmitted to a simple boot monitor on the target over a serial link. Tools to do this are usually supplied with your development environment. Consult the documentation on

your target platform and development toolchain for details of how to program applications into non-volatile memory.

All ports of RTA-OS3.0 are ROMable and are tested running on a target CPU without any debugger or development equipment connected.

4 Finding out more

Your RTA-OS3.0 distribution includes the following manuals:

`<install dir>\Documents`

***RTA-OS3.0 Getting Started Guide*.** The document that you are reading now. This guide explains how to install the product and describes the underlying principles of the operating system.

***RTA-OS3.0 Release Note*.** This document provides information about the release, including a list of changes from previous releases and a list of known issues.

***RTA-OS3.0 User Guide*.** This guide explains the concepts behind AUTOSAR OS R3.0 and shows you how to use RTA-OS3.0 to configure the OS and integrate it into your application

***RTA-OS3.0 Reference Guide*.** This guide provides a complete reference to the API and programming conventions for RTA-OS3.0.

`<install dir>\Targets\VRTA_n.n.n`

***RTA-OS3.0 VRTA Port Guide*.** This guide explains implememtation-specific details for the VRTA port plug-in.

***RTA-OS3.0 VRTA Release Note*.** This document provides information about the VRTA port plug-in release, including a list of changes from previous releases and a list of known issues.

***RTA-OS3.0 Virtual ECU User Guide*.** This guide explains how to use the Virtual ECU environment included with the VRTA port plug-in.

`<install dir>\Targets\<TargetCompiler>_n.n.n`

***RTA-OS3.0 Target/Compiler Port Guide*.** Each port of RTA-OS3.0 is supplied with a Port Guide. The Port Guide tells you specific information about the interaction between RTA-OS3.0, your toolchain and your target hardware. For example, valid compiler options, register settings, interrupt handling etc. The Port Guide also gives performance and resource usage information for the OS.

***RTA-OS3.0 Target/Compiler Release Note*.** This document provides information about the port plug-in release, including a list of changes from previous releases and a list of known issues.

4.1 Related Reading

4.1.1 OSEK

OSEK/VDX Operating System (Version 2.2.3 17th February 2005) is the OSEK OS specification and describes the behavior of the OS in detail.

OSEK/VDX Binding Specification (Version 1.4.2 15th July 2004) describes how OSEK OS inter-operates with other OSEK standards.

OSEK RunTime Interface (ORTI) (Part A: Language Specification Version 2.2 14th November 2005 and Part B: OSEK Objects and Attributes Version 2.2 25th November 2005) defines the core ORTI debugging language and the standard attributes supported by OSEK OS.

4.1.2 AUTOSAR

AUTOSAR SWS OS (R3.0 V3.0.0 Rev 0001 7.Dec.2007) is the AUTOSAR OS specification and describes AUTOSAR's extensions to OSEK OS in detail.

AUTOSAR SWS GPT Driver (R3.0 V2.2.0 Rev 0001 11.Dec.2007) explains the AUTOSAR "General Purpose Timer" (Gpt) driver that the OS uses[1]

AUTOSAR SWS RTE (R3.0 V2.0.0 Rev 0001 20.Dec.2007) describes the AUTOSAR RTE. This is required reading if you want to know how an AUTOSAR RTE interacts with an AUTOSAR OS.

AUTOSAR SRS General (R3.0 V2.2.0 Rev0001 10.Dec.2007) describes the general properties that apply to all AUTOSAR basic software modules. Note that the AUTOSAR OS document defines specific deviations from this general SRS that apply to the OS

AUTOSAR SWS StandardTypes (R3.0 V1.2.0 Rev 0001 29.Nov.2007) defines the standard AUTOSAR types and the include structure for basic software modules like the OS.

AUTOSAR SWS PlatformTypes (R3.0 V2.2.0 Rev 0001 13.Nov.2007) describes how target platform characteristics are abstracted in AUTOSAR.

AUTOSAR SWS MemoryMapping (R3.0 V1.1.0 Rev 0001 12.Dec.2007) describes how AUTOSAR handles memory mapping of software. RTA-OS3.0 uses this concept to place parts of the OS in user nominated sections.

[1] Note - while the AUTOSAR OS specification requires that the OS uses the Gpt, in practice the Gpt is not sufficiently powerful enough to be of any practical use. RTA-OS3.0 therefore does not make use of the Gpt driver directly. Please consult the *RTA-OS3.0 Release Note* for further details.

AUTOSAR SWS CompilerAbstraction (R3.0 V2.0.0 Rev 0001 27.Nov.2007) describes how compiler properties such as type definitions, pointers, access to near and far memory etc. are abstracted in AUTOSAR. RTA-OS3.0 uses this concept to ensure that all internal OS code conforms to the AUTOSAR standard.

5 Contacting ETAS

5.1 Technical Support

Technical support is available to all RTA-OS3.0 users with a valid support contract. If you do not have such a contract then please contact ETAS through one of the addresses listed in Section 5.2.

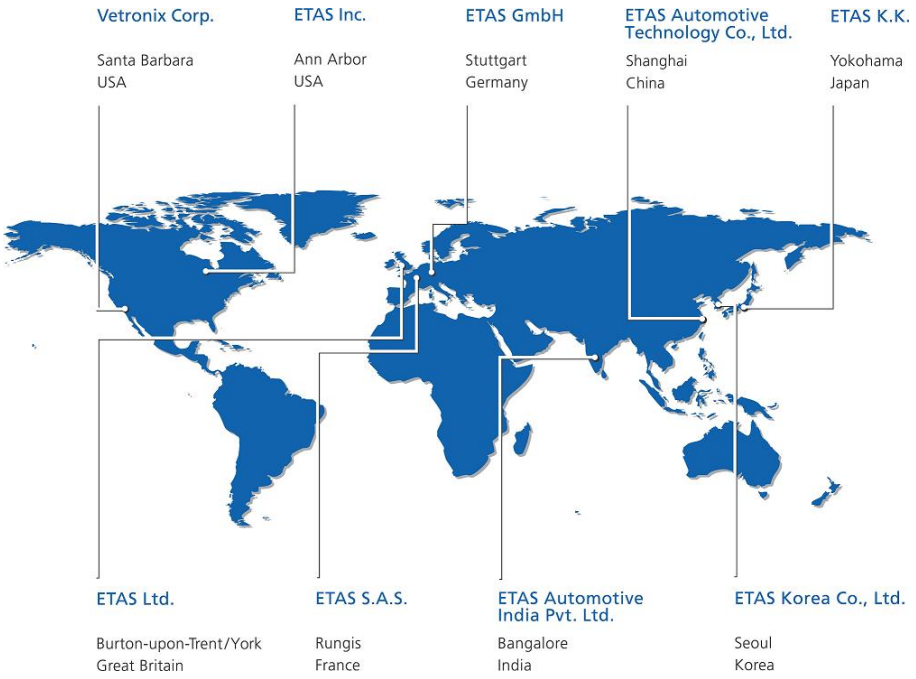
The best way to get technical support is by email. Any problems or questions should be sent to: rta.hotline.uk@etas.com

It is helpful if you can provide support with the following information:

- your support contract number.
- your .xml/.rtaos configuration files.
- the error message you received and the file `Diagnostic.dmp` if it was generated.
- the command line that results in an error message.
- the version of the ETAS tools you are using.
- the version of your compiler tool chain you are using.

If you prefer to discuss your problem with the technical support team you can contact them by telephone during normal office hours (0900-1730 GMT/BST). The telephone number for the RTA-OS3.0 support hotline is: +44 (0)1904 562624.

5.2 General Enquiries



Europe

Excluding France, Belgium, Luxembourg, United Kingdom and Scandinavia

ETAS GmbH

Borsigstrasse 14	Phone:	+49 711 89661-0
70469 Stuttgart	Fax:	+49 711 89661-300
Germany	E-mail:	sales.de@etas.com
	WWW:	www.etas.com

France, Belgium and Luxemburg

ETAS S.A.S.

1, place des États-Unis	Phone:	+33 1 56 70 00 50
SILIC 307	Fax:	+33 1 56 70 00 51
94588 Rungis Cedex	E-mail:	sales.fr@etas.com
France	WWW:	www.etas.com

United Kingdom and Scandinavia

ETAS Ltd.

Studio 3, Waterside Court	Phone:	+44 1283 54 65 12
Third Avenue, Centrum 100	Fax:	+44 1283 54 87 67
Burton-upon-Trent	E-mail:	sales.uk@etas.com
Staffordshire DE14 2WQ	WWW:	www.etas.com
United Kingdom		

USA

ETAS Inc.		
3021 Miller Road	Phone:	+1 888 ETAS INC
Ann Arbor	Fax:	+1 734 997-9449
MI 48103	E-mail:	sales.us@etas.com
USA	WWW:	www.etas.com

Japan

ETAS K.K.		
Queen's Tower C-17F	Phone:	+81 45 222-0900
2-3-5, Minatomirai, Nishi-ku	Fax:	+81 45 222-0956
Yokohama 220-6217	E-mail:	sales.jp@etas.com
Japan	WWW:	www.etas.com

Korea

ETAS Korea Co. Ltd.		
4F, 705 Bldg. 70-5	Phone:	+82 2 5747-016
Yangjae-dong, Seocho-gu	Fax:	+82 2 5747-120
Seoul 137-889	E-mail:	sales.kr@etas.com
Korea	WWW:	www.etas.com

P.R.China

ETAS (Shanghai) Co., Ltd.		
2404 Bank of China Tower	Phone:	+86 21 5037 2220
200 Yincheng Road Central	Fax:	+86 21 5037 2221
Shanghai 200120	E-mail:	sales.cn@etas.com
P.R. China	WWW:	www.etas.com

India

ETAS Automotive India Pvt. Ltd.		
No. 690, Gold Hill Square, 12F	Phone:	+91 80 4191 2585
Hosur Road, Bommanahalli	Fax:	+91 80 4191 2586
Bangalore, 560 068	E-mail:	sales.in@etas.com
India	WWW:	www.etas.com

Index