



# RTI

*The Run-Time Infrastructure  
for HLA*



*MÄK Technologies  
185 Alewife Brook Parkway  
Cambridge, MA 02138 USA*

Copyright 1998, MÄK Technologies  
All rights Reserved. Printed in the United States.

Second edition: October 1998

Under copyright laws, no part of this document may be copied or reproduced in  
any form without prior written consent of MÄK Technologies, Inc.

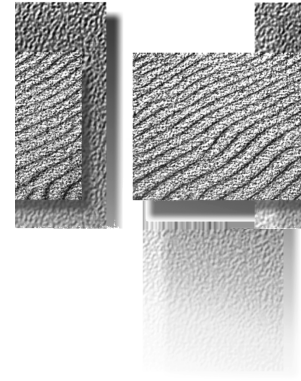
VR-Link, MÄK Logger, MÄK Dial-a-Sim, MÄK Plan View Display, and MÄK Stealth  
are trademarks of MÄK Technologies, Inc.  
All other trademarks are owned by their respective companies.

**MÄK Technologies**  
185 Alewife Brook Parkway  
Cambridge, MA 02138 USA

Voice: 617-876-8085  
Fax: 617-876-9208

[info@mak.com](mailto:info@mak.com)  
[www.mak.com](http://www.mak.com)

Revision RTI-1.3-1-981008



# *Contents*

---

## **Preface**

---

|                                          |      |
|------------------------------------------|------|
| MÄK Products for the Virtual World ..... | vi   |
| Getting Technical Support .....          | vii  |
| Document Conventions .....               | viii |

## **Chapter 1 Introduction to the MÄK RTI**

---

|                                                |     |
|------------------------------------------------|-----|
| 1.1. About RTIs .....                          | 1-2 |
| 1.2. Compatibility with the DMSO RTI .....     | 1-2 |
| 1.2.1. Compiling Applications .....            | 1-2 |
| 1.2.2. Dynamic Linking .....                   | 1-3 |
| 1.2.3. Static Linking .....                    | 1-3 |
| 1.2.4. The FED File .....                      | 1-3 |
| 1.2.5. The RID File .....                      | 1-3 |
| 1.2.6. Network Compatibility .....             | 1-3 |
| 1.3. Documentation .....                       | 1-4 |
| 1.4. Compiling and Linking .....               | 1-4 |
| 1.5. Running Applications .....                | 1-5 |
| 1.5.1. Reading The FED File .....              | 1-5 |
| 1.6. Services Implemented .....                | 1-6 |
| 1.7. Accessing the RTI's File Descriptor ..... | 1-6 |
| 1.8. rtidump .....                             | 1-7 |
| 1.9. Packet Server (UNIX Only) .....           | 1-7 |

## **Chapter 2 Installing and Configuring the MÄK RTI**

---

|                                                         |     |
|---------------------------------------------------------|-----|
| 2.1. System Requirements .....                          | 2-2 |
| 2.2. Installing the MÄK RTI .....                       | 2-2 |
| 2.2.1. Configuring Your System to Use the MÄK RTI ..... | 2-2 |
| 2.2.2. Running Applications with the MÄK RTI .....      | 2-3 |
| 2.3. The rtiexec .....                                  | 2-3 |

## Contents

---

|                                                            |      |
|------------------------------------------------------------|------|
| 2.4. The RID File .....                                    | 2-4  |
| 2.5. The FED File .....                                    | 2-5  |
| 2.6. Setting Up and Using the FLEXlm License Manager ..... | 2-6  |
| 2.6.1. Setting up the License Server .....                 | 2-6  |
| 2.6.2. Running the License Server .....                    | 2-9  |
| 2.6.3. Troubleshooting the FLEXlm License Manager .....    | 2-10 |

## Index



## *Preface*

---

This manual is for developers of HLA simulation applications and for persons who need to install or use the MÄK RTI.

This manual assumes that you are familiar with basic UNIX or DOS operations and that you know how to work in your computer environment, either OSF/Motif, or Windows.

The MÄK RTI is a dynamic product, constantly evolving to meet user needs. See the release notes for information about improvements or new features that have been added to the RTI since this manual went to press.

This manual is organized as follows:

Chapter 1, *Introduction to the MÄK RTI* describes the MÄK RTI and its features.

Chapter 2, *Installing and Configuring the MÄK RTI* describes installation and configuration tasks.

## ***MÄK Products for the Virtual World***

The RTI is a member of the MÄK Technologies line of HLA/DIS software products designed to streamline the process of developing and using networked simulated environments.

In addition to the RTI, the MÄK Technologies product line includes:

- ♦ The VR-Link Network Toolkit. The Toolkit is an object oriented library of C++ functions and definitions that implement the HLA RPR FOM and the DIS protocol. This library minimizes the time and effort required to build and maintain new HLA/DIS-compliant applications, and to integrate such compliance into existing applications.

The Toolkit includes a set of sample debugging applications and their source code. The source code serves as an example of how to use the VR-Link Toolkit to write applications. The executables provide valuable debugging services such as generating a predictable stream of HLA or DIS messages, and displaying the contents of messages transmitted on the network.

VR-Link is the basis of the other MÄK products.

- ♦ The MÄK Stealth. The Stealth displays a realistic, three-dimensional view of your virtual world. You can view this world from the inside of a simulated moving vehicle, or place the eyepoint at another moving or stationary location. The Stealth lets you switch rapidly among several predefined viewpoints while the simulation is underway.
- ♦ The MÄK Logger. The *Logger*, also called the *Data Logger*, can record HLA and DIS exercises and play them back for after-action review. You can play a recorded file at speeds above or below normal, or in reverse, and locate areas of interest quickly. The Logger lets you choose between a graphical interface, or a shell-style, command-line interface.

The Logger comes with a set of utility programs that let you inspect, combine and edit Logger recordings.

- ♦ The MÄK Plan View Display. The Plan View Display (PVD) provides a two-dimensional, “big picture” of a virtual environment. It shows simulated entities, such as vehicles, soldiers, and tanks moving over a 2D-map display. The MÄK PVD is compatible with both Distributed Interactive Simulation (DIS) and High Level Architecture (HLA) simulation standards.
- ♦ Dial-a-Sim. Dial-a-Sim is a reconfigurable DIS simulator program that lets you create and simulate vehicles quickly, with or without programming. Dial-a-Sim’s graphical interface lets you assemble vehicles by choosing from a collection of predefined components. You can adjust the physical properties and positioning of these components to meet your exact needs.

Once your vehicle is complete, you can drive it in a networked simulation. Dial-a-Sim provides a set of on-screen controls, and supports PC-compatible steering wheel assemblies and other hardware controls.

For more information about these products, contact your sales representative.

## ***Getting Technical Support***

For RTI technical support, information about upgrades, and information about other MÄK products, you can reach us in the following ways:

|                                              |                                                                          |
|----------------------------------------------|--------------------------------------------------------------------------|
| For sales and upgrade information by e-mail: | <a href="mailto:vrlink-info-sales@mak.com">vrlink-info-sales@mak.com</a> |
| For technical support by e-mail:             | <a href="mailto:vrlink-tech-spt@mak.com">vrlink-tech-spt@mak.com</a>     |
| For general MÄK information via the Web:     | <a href="http://www.mak.com">http://www.mak.com</a>                      |

|                                |                                                                      |
|--------------------------------|----------------------------------------------------------------------|
| Send postal correspondence to: | MÄK Technologies<br>185 Alewife Brook Parkway<br>Cambridge, MA 02138 |
|--------------------------------|----------------------------------------------------------------------|

|                    |              |              |
|--------------------|--------------|--------------|
| Call or fax us at: | Fax:         | 617-876-9208 |
|                    | Voice Phone: | 617-876-8085 |

## ***Document Conventions***

This document uses the following typographic conventions:

**Monospaced Type**      Indicates commands or values you enter.

*Monospaced Italic*      Indicates variables that you replace with appropriate values.

[ ]      Indicates optional arguments.

|      Separates options in a command where only one option may be chosen at a time.

/      Indicates a directory. Since MÄK products run on both UNIX and PC platforms, we use the / (slash) for generic discussions of pathnames. If you are running on a PC, substitute a \ (backslash) when you type pathnames.

**Bold type**      Indicates a new term.

*Italic*      Indicates a file name or path, or a class name.

sansSerif      Indicates a parameter or argument.

>      Indicates a one-step procedure.

Menu→Option      Indicates a menu choice. For example, an instruction to select the Save option from the File menu appears as:  
Choose File→Save.

Directory names are preceded with dot and slash characters that show their position with respect to the RTI home directory. For example, the directory *rti/bin* appears in the text as *./bin*.





# *Introduction to the MÄK RTI*

---

The MÄK RTI is an implementation of the Run-Time Infrastructure portion of the High-Level Architecture (HLA). The MÄK RTI implements a subset of the HLA Interface Specification. Refer to the *MÄK RTI Release Notes* for information about current version information.

This chapter includes the following:

- ♦ About RTIs, on page 1-2
- ♦ Compatibility with the DMSO RTI, on page 1-2
- ♦ Documentation, on page 1-4
- ♦ Compiling and Linking, on page 1-4
- ♦ Running Applications, on page 1-5
- ♦ Services Implemented, on page 1-6
- ♦ Accessing the RTI's File Descriptor, on page 1-6
- ♦ rtidump, on page 1-7
- ♦ Packet Server (UNIX Only), on page 1-7

## **1.1. About RTIs**

An RTI (Run-Time Infrastructure) is a software library (and perhaps supporting executables) that implements the HLA Interface Specification. In HLA, applications exchange FOM data through RTI calls, which means that all HLA applications need to use an RTI. (Applications that use VR-Link typically do not need to make RTI calls directly, but only because VR-Link makes the RTI calls for them.)

Because of differences in the low-level network mechanisms used by different RTI implementations (which include, but are not limited to packet layout), applications that want to interoperate in the same federation execution must use the same RTI implementation. Because RTIs are provided as dynamic libraries that implement a fixed API, a federation can easily switch from one RTI implementation to another between runs (without even recompiling the applications), but during each run, all participants must agree on which RTI to use, much as they must also agree on which FOM to use.

The two RTI implementations against which all MÄK products have been tested are the MÄK RTI (which comes with all MÄK products), and the DMSO RTI (which is available from the DMSO web site).

## **1.2. Compatibility with the DMSO RTI**

With the exception of one additional function (see “Accessing the RTI’s File Descriptor”, on page 1-6), the MÄK RTI has the exact same API as the DMSO RTI. The MÄK RTI also uses the exact same header files as those supplied with the DMSO RTI. This section discusses the following aspects of compatibility:

- ♦ Compiling Applications
- ♦ Dynamic Linking
- ♦ Static Linking
- ♦ The FED File
- ♦ The RID File
- ♦ Network Compatibility

### **1.2.1. Compiling Applications**

An application built against the MÄK RTI does not need to be recompiled in order to be used with the DMSO RTI, or vice versa. For more information, see “Compiling and Linking”, on page 1-4.

### 1.2.2. Dynamic Linking

Because the RTI is typically accessed by an application through a dynamic library (called *libRTI.so* on UNIX platforms and *rti.dll* on PCs), applications do not need to be relinked in order to switch between the MÄK RTI and the DMSO RTI. Applications built using the DMSO RTI should run with the MÄK RTI unmodified, and vice versa. All that is required is that the appropriate version of *libRTI.so* or *rti.dll* be located within the shared library search path. For more information, see “Compiling and Linking”, on page 1-4.

### 1.2.3. Static Linking

The MÄK RTI contains a static version of the library (*libRTI.a*). If you link statically with this library, you must relink to switch to the DMSO RTI.

### 1.2.4. The FED File

The MÄK RTI uses the same FED file format as the DMSO RTI. (The FED file is an RTI-readable description of the relevant information in the FOM). It is not necessary to modify your FOM or FED file to switch between the two. See “The FED File”, on page 2-5 for details.

### 1.2.5. The RID File

The MÄK RTI uses a different RID file format than the DMSO RTI. The RID (RTI-Initialization Data) file provides RTI-implementation-specific configuration parameters, and hence cannot be shared among different RTIs. A MÄK RTI RID file is an *.mtl* file with the same format as the files used to configure other MÄK products. See “The RID File”, on page 2-4 for details.

**Note:** The RID file is not required. Use it if you want to change the default parameters.

### 1.2.6. Network Compatibility

The MÄK RTI is not network compatible with the DMSO RTI. (In general, any two different RTI implementations will not be network compatible.) This means that if a set of applications want to play together in the same federation execution, they must all be using the same RTI implementation. (But, of course, they can all then switch to another RTI implementation and play in another federation execution).

### 1.3. Documentation

Because the MÄK RTI implements the same interface specification as the DMSO RTI, the documentation provided with the DMSO RTI applies to the MÄK RTI as well, apart from the few differences we describe in this manual. The DMSO documentation is included with the MÄK RTI.

### 1.4. Compiling and Linking

To build an application that uses the MÄK RTI, you must specify the RTI's *include* directory on your compile line, and the RTI's *lib* directory and the name of the RTI library on your link line.

For example, if you have installed the RTI in */usr/products/makRtix.x*, the following should appear on your compile line:

```
-I/usr/products/makRtix.x/include
```

and the following should appear on your link line:

```
-L/usr/products/makRtix.x/lib -lRTI
```

The MÄK RTI uses *libm*, the standard system math library. Therefore the flag `-lm` must appear on your link line after `-lRTI`.

#### Linking Under Solaris

Under Solaris, you also need the following flags on your link line:

- ♦ `-lsocket`
- ♦ `-lnsl`
- ♦ `-lint`

because the MÄK RTI uses these system libraries.

## 1.5. Running Applications

Before you run any MÄK-RTI-based applications, the license server must be running on a single machine designated as the license server. In addition, you must set the `LM_LICENSE_FILE` environment variable, as described in “Setting Up and Using the FLEXlm License Manager”, on page 2-6, on each machine on which you are running an application that uses the MÄK RTI.

If you have linked dynamically with the *libRTI.so* or *rti.dll* library, the library needs to be accessible to the run-time linker, like any other shared library.

- > To make the library accessible to the run-time linker, include the RTI's *lib* directory in the dynamic library search path, which is configurable through an environment variable on most systems.

Table 1-1 lists the environment variables for the systems supported by the MÄK RTI.

**Table 1-1:** Environment variables

| System        | Environment Variable |
|---------------|----------------------|
| IRIX6 N32 ABI | LD_LIBRARYN32_PATH   |
| IRIX6 O32 ABI | LD_LIBRARY_PATH      |
| SunOS4        | LD_LIBRARY_PATH      |
| Solaris       | LD_LIBRARY_PATH      |
| Dec Alpha     | PATH                 |
| Win NT/95     | PATH                 |

As an alternative, you can copy the library to the standard place that shared libraries are stored on your system (typically in */usr/lib* on UNIX machines, or in *\winnt\system32* or *\windows\system* on PCs).

### 1.5.1. Reading The FED File

In the RTI 1.3 API, the FED file name is passed to the `createFederationExecution()` service along with the federation execution name. When `joinFederationExecution()` is called with a federation execution name, the FED file that was associated with the execution name in an earlier create call is read.

In the MAK RTI, there is not necessarily a central *rtiexec* server that manages the list of associations between execution names and FED file names. Therefore, all federates should call `createFederationExecution()` before calling `joinFederationExecution()` (in order to tell us what FED file to use) even if you know that the federation execution has already been created by another federate.

## 1.6. Services Implemented

MÄK RTI implements a subset of the RTI services. We have implemented the services that are most likely to be needed by the real-time platform-level simulation community. Refer to the *MÄK RTI Release Notes* for a list of services implemented in the latest release.

The MÄK RTI implements only the best-effort communications mechanism, and does not publish any MOM data.

## 1.7. Accessing the RTI's File Descriptor

The MÄK RTI adds one function to the API provided by the DMSO RTI. The function is called `RtReadFileDescriptor()` and is declared in *makRti.h*. This function returns a file descriptor on which data is guaranteed to be present whenever there is input from the network that the RTI needs to process.

This file descriptor can be passed to the system call `select` (or to something like the X function `XtAppAddInput()`) in applications that want to yield the CPU until there is HLA input pending, avoiding unnecessary polling *using* `RTI::tick()`. When the application regains control, `RTI::tick()` should be called so that the pending input can be processed by the RTI.

`RtReadFileDescriptor()` has the following prototype:

```
int RtReadFileDescriptor(RTI::RTIambassador* amb);
```

**Note:** If you use this extension to the standard RTI API, your application will no longer be link-compatible with the DMSO RTI, as it does not implement this function. If your application needs to switch between RTI implementations, you can place the calls within `#ifdefs`.

## 1.8. *rtidump*

The MÄK RTI includes a debugging utility called *rtidump*, that can passively listen to a federation execution and print out all messages being exchanged by the RTI on the network.

*rtidump* must be configured to listen to the port and address being used for the desired federation execution.

The syntax for *rtidump* is:

```
rtidump -A address -P port
```

where *port* and *address* are the port and address to use for the federation execution.

- > To configure *rtidump* to listen to the port and address being used for a federation execution, use *rid.mtl*, or the *-P* and *-A* command-line options.

Command line options take precedence over the values in *rid.mtl*.

- > To quit *rtidump*, press “q”, then return.

## 1.9. *Packet Server (UNIX Only)*

On UNIX systems, you can off-load the MÄK RTI's reading of packets from the network to its own asynchronous process using the MÄK Packet Server - a separate product available from MÄK. This is particularly useful on multiprocessor machines, and in cases where you're running multiple federates on the same machine. In the latter case, a packet would be read from the network only once by the packet server, and viewed by each federate through shared memory.







## ***Installing and Configuring the MÄK RTI***

When you install a MÄK product, you have the option of installing the MÄK RTI. This chapter provides details about the installation and additional configuration that you need to do.

Topics include:

- ♦ System Requirements, on page 2-2
- ♦ Installing the MÄK RTI, on page 2-2
- ♦ Setting Up and Using the FLEXlm License Manager, on page 2-6
- ♦ The rtiexec, on page 2-3
- ♦ The RID File, on page 2-4
- ♦ The FED File, on page 2-5

## 2.1. System Requirements

The MÄK RTI is a dynamic product that is responsive to the rapidly changing HLA protocol. Please refer to the Release Notes that accompany your version of the MÄK RTI to see the latest information about system requirements and release levels.

## 2.2. Installing the MÄK RTI

The MÄK RTI is included with all MÄK products. When you install a MÄK product, you have the option of installing the MÄK RTI as part of the installation process.

### UNIX

The installation process creates a link called *RTI* in each product's root directory, which points to the *makRti* directory, in which the MÄK RTI software is located.

### PCs

The project files point to the default installation directory for the RTI.

### 2.2.1. Configuring Your System to Use the MÄK RTI

The RTI dynamic library needs to be located somewhere on your dynamic library search path. This path is indicated by an environment variable as listed in Table 2-1.

**Table 2-1:** Environment variables

| System        | Environment Variable |
|---------------|----------------------|
| IRIX6 N32 ABI | LD_LIBRARYN32_PATH   |
| IRIX6 O32 ABI | LD_LIBRARY_PATH      |
| SunOS4        | LD_LIBRARY_PATH      |
| Solaris       | LD_LIBRARY_PATH      |
| Dec Alpha     | PATH                 |
| Win NT/95     | PATH                 |

The MÄK RTI needs to know where to find the following configuration files:

- ♦ The FED file (*.fed*) for your federation execution (required)
- ♦ The RID file (*rid.mtl*) (optional)

Put the configuration files in the directory from which you are running, or set the environment variable RTI\_CONFIG to the directory that contains them.

### 2.2.2. Running Applications with the MÄK RTI

You do not need an RTI server or central application such as *rtiexec* to use the MÄK RTI. However, be sure the license server is running, then start the applications, and they should be able to communicate.

## 2.3. The *rtiexec*

No RTI server or other central application is necessary in order to use the MÄK RTI. Just start the applications, and they should be able to communicate, as long as they are all configured to use the same multicast or broadcast address and UDP port (see “The RID File”, on page 2-4).

However, you can use an *rtiexec* application to keep track of the current set of valid federation executions, and to guarantee that federate IDs assigned to federates joining a federation execution are unique.

If you are not using the *rtiexec*:

- The `createFederationExecution()` service returns successfully even if the named execution has already been created
- The `destroyFederationExecution()` service always returns successfully even if the named execution does not currently exist
- The `joinFederationExecution()` service returns successfully even if the named federation execution has never been created using `createFederationExecution()`

In addition, the federate ID returned by the `joinFederationExecution()` service is not guaranteed to be unique when not using the *rtiexec*. It is computed by taking the federate’s process ID modulo 10,000. In an execution with ten federates running on ten different machines, there is about a one in 200 chance that two federates will be assigned the same federate ID.

- > To use the *rtiexec*, run *rtiexec* from the MÄK RTI’s *bin* directory, making sure that it is configured to use the same broadcast or multicast address and port number as all of the federates. (See “The RID File”, on page 2-4).
- > To override the port number in the *RID* file use *rtiexec*’s `-P` option.
- > To override the IP address use the `-A` option.

For example:

```
rtiexec -P 4444 -A 225.9.9.9
```

You must indicate to the federates that they should try to connect to the *rtiexec*.

- > To indicate to the federates that they should try to connect to the *rtiexec*, set the `RTI_useRtiExec` parameter to 1 in the RID file.

If you are using the *rtiexec* in a particular federation execution, it should remain running until that federation execution has been destroyed.

- > To exit the *rtiexec*, type q, return, in its console.
- > To inhibit the output of diagnostics, use the -q option.

This is often desirable when running *rtiexec* in the background.

## 2.4. The RID File

When an application creates an instance of *RTI::RTIambassador*, the RTI can load a RID file that contains various configuration parameters. This file is called *rid.mtl*. The RTI looks for this file first in the current directory (*.*), then in the directory indicated by the environment variable *RTI\_CONFIG*. If no *rid.mtl* file is found, or if some parameters do not appear in the file, default parameter values are used.

A MÄK RTI RID file is an *.mtl* file that uses the same format as the files used to configure other MÄK products. It contains a line for each parameter that you would like to set, in the format:

```
(setqb parameterName value)
```

Table 2-2 lists the supported parameters.

**Table 2-2:** RID file parameters (Sheet 1 of 2)

| Parameter                 | Default                            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------------------|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>RTI_useRtiExec</i>     | 0<br>(do not use <i>rtiexec</i> )  | Determines whether the application should attempt to connect to an <i>rtiexec</i> application when it creates, destroys, or joins a federation execution. Valid values are 0 and 1.                                                                                                                                                                                                                                                                                            |
| <i>RTI_udpPort</i>        | 4000                               | Indicates the UDP port number to be used for messages sent among federates, as well as between federates and the <i>rtiexec</i> if applicable. All federates in a federation execution must use the same value.                                                                                                                                                                                                                                                                |
| <i>RTI_destAddrString</i> | 229.7.7.7<br>(a multicast address) | Indicates the IP address to which federation execution traffic is sent. This is typically a broadcast address or multicast address. All federates in a federation execution must use the same value. A value of "0.0.0.0" is a special value that indicates that the <i>default_broadcast_address</i> should be used. If any of your federates are running on platforms on which multicast is not supported, you must use a broadcast address rather than a multicast address. |

Table 2-2: RID file parameters (Sheet 2 of 2)

| Parameter       | Default | Description                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RTI_checkFlag   | 1       | Indicates whether the RTI should check the validity of attributes and parameters and the ownership of attributes passed to updateAttributeValues() and sendInteraction(). Setting this flag to 0 results in a small performance gain, but it means that the AttributeNotDefined, AttributeNotOwned, and InteractionParameterNotDefined exceptions will never be thrown by these services. Valid values are 0 and 1. |
| RTI_notifyLevel | 2       | Indicates the categories of notification messages to print. Values are:<br>0 - Fatal<br>1 - Warn<br>2 - Notify<br>3 - Verbose                                                                                                                                                                                                                                                                                       |

A sample *rid.mtl* file is provided in the MÄK RTI root directory.

## 2.5. The FED File

The MÄK RTI uses the same FED file format as the DMSO RTI. When the joinFederationExecution() service is invoked, the RTI reads the FED file passed to the create call for the named execution. The MÄK RTI first looks in the current directory (./), and then in the directory indicated by the RTI\_CONFIG environment variable. All federates in a federation execution must use identical FED files.

## 2.6. Setting Up and Using the FLEXlm License Manager

The RTI, like other MÄK products, uses the FLEXlm license manager. Before you can use the RTI, you must have a license server installed and a valid *license.dat* file. (For information about FLEXlm, view the FAQ page at [www.globetrotter.com/flmfaq.htm](http://www.globetrotter.com/flmfaq.htm).) Contact the MÄK Technologies sales department for information about special licensing agreements.

### 2.6.1. Setting up the License Server

When you install the RTI, FLEXlm files are installed in the *./flexlm* directory. The procedure you need to follow to set up license management depends on whether or not you have previously installed a MÄK product that uses the license manager.

#### If you have no other MÄK products installed

The machine on which you run the license manager is the license server. It can be the same machine as the one on which you installed the RTI, or a different one. If you want to use a different machine as the license server, copy the files in the *./flexlm* directory to that machine.

To set up the FLEXlm license server:

1. Change to the *flexlm* directory on the license server.
2. Execute *lmhostid*. This program generates a unique number that MÄK uses to generate your license activation codes.
3. Call or e-mail MÄK with the name of the server and the number obtained by *lmhostid*. MÄK will e-mail or FTP you a file named *license.dat*.
4. Put the *license.dat* file in the *flexlm* directory on the license server and on all machines on which you have the RTI installed.
5. On each machine that has a *license.dat* file, set the environment variable LM\_LICENSE\_FILE to the absolute path to *license.dat*.

#### Windows 95

On Windows 95, set environment variables in the *Autoexec.bat* file, for example:

```
set LM_LICENSE_FILE="C:\Program Files\MÄK Technologies\  
flexlm\license.dat"
```

**Note:** Do not put spaces around the = sign. Put quotes around the path.

### **Windows NT**

On Windows NT, you set environment variables in the Environment tab of the Properties dialog box for the machine:

1. Right-click the My Computer icon on the Desktop.
2. On the pop-up menu, select Properties. The System Properties dialog box opens.
3. Select the Environment tab.
4. In the Variable text field, type LM\_LICENSE\_FILE.
5. In the Value field, type the pathname of the file, for example:

`"C:\Program Files\MÄK Technologies\flexlm\license.dat"`

### **UNIX**

On UNIX, set the variable similarly to the following example:

```
setenv LM_LICENSE_FILE /usr/local/rti/flexlm/license.dat
```

6. Edit the *license.dat* file to specify the location of the *maklmgrd* executable on the server machine. The file supplied by MÄK represents this location with a dot as shown below:

```
DAEMON maklmgrd .
```

Replace the dot with your absolute path to *maklmgrd*, for example:

```
DAEMON maklmgrd "C:\Program Files\MÄK Technologies\flexlm\maklmgrd"
```

**Note:** On PCs, you must enclose the path in quotes.

### **If you already have a license.dat file from other MÄK products**

If you have other MÄK products, you can set up a separate license server for the RTI, in which case follow the procedure on the previous page. Or, you can merge the license information for the RTI with that of previous products and use the same license server.

To obtain and merge license information:

1. Change to the *flexlm* directory on the license server.
2. Execute *lmhostid*. This program generates a unique number that MÄK uses to generate your license activation codes.
3. Call or e-mail MÄK with the name of the server, the number obtained by *lmhostid*. MÄK will e-mail or FTP you a file named *license.dat*. The file is in a format similar to the following:

```
SERVER mollusc 006097752f93 4567
DAEMON maklmgrd .
PACKAGE rti maklmgrd 1999.177 60D00041FA200D8A795D COMPONENTS=rti1
INCREMENT rti maklmgrd 1999.177 1-jan-0 1 0BB480511615C175B8CF \
    PLATFORMS=i86_n
```

Your existing license.dat file is similar to the following:

```
SERVER mollusc 006097752f93 4567
DAEMON maklmgrd "c:\Program Files\MÄK Technologies\stealth\flexlm\
    maklmgrd"
PACKAGE makproduct maklmgrd 1999.177 40C020D12D609AB04D55 \
    COMPONENTS="hprod1 dprod1 prod1"
INCREMENT makproduct maklmgrd 1999.177 1-jan-0 1 AB4450916DD1944435D2 \
    PLATFORMS=i86_n
```

**Note:** The line continuation marks (\) in the examples are for clarity of presentation. They may not be present in the files you receive.

4. Open up the two files in a text editor. Append everything except the Server and Daemon lines (the Server lines will be identical) from the new *license.dat* file to the end of the old *license.dat* file. For example:

```
SERVER mollusc 006097752f93 4567
DAEMON maklmgrd "c:\Program Files\MÄK Technologies\stealth\flexlm\
    maklmgrd"
PACKAGE makproduct maklmgrd 1999.177 40C020D12D609AB04D55 \
    COMPONENTS="hprod1 dprod1 prod1"
INCREMENT makproduct maklmgrd 1999.177 1-jan-0 1 AB4450916DD1944435D2 \
    PLATFORMS=i86_n
PACKAGE rti maklmgrd 1999.177 60D00041FA200D8A795D \
    COMPONENTS=rti1
INCREMENT rti maklmgrd 1999.177 1-jan-0 1 0BB480511615C175B8CF \
    PLATFORMS=i86_n
```

5. Put the updated *license.dat* file in the *flexlm* directory on the license server and on all machines on which you have the RTI installed.



### **2.6.2. Running the License Server**

The FLEXlm license server must be running on the server machine and the LM\_LICENSE\_FILE variable must be set on the local machine before you can run your MÄK applications.

- > To start the license server, on the server machine only, execute *runLm* from the *./flexlm* directory.
- > To obtain the current status of the server, run *lmstat*.

#### **Shutting Down the License Server**

- > To force all applications to check in their licenses and shut down the license server, run *lmdown*.

### 2.6.3. Troubleshooting the FLEXlm License Manager

For general troubleshooting information, refer to the FLEXlm FAQ at [www.globe-trotter.com/flmfaq.htm](http://www.globe-trotter.com/flmfaq.htm). It describes common problems, including the majority that have been reported to MÄK support engineers. Please consult the FAQ first when you have a problem.

#### Unable to Get A License - Port in Use

If you are unable to get a license for a MÄK application and the FLEXlm log file indicates that the port is in use, it is possible that you have more than one *lmgrd* or *maklmgrd* processes running. It is also possible that an application that was using a license is thought to have terminated, but is still alive. To verify and resolve these possibilities:

1. See if more than one *lmgrd* or *maklmgrd* process is running. On Windows NT, check the Task Manager. On UNIX, enter the following command:  

```
ps -aux | grep lmgrd
```

```
ps -aux | grep maklmgrd
```
2. If more than one of either process is running, note the process ids (PID) for both executables and kill all of them, for example:

```
kill -9 1385 878
```

where 1385 and 878 are process IDs.

It is important that you kill all the processes to ensure a clean restart for the license server.

3. See if there are old processes running that may be using a license. On Windows NT, check the task manager. On UNIX, use `ps -aux`.

**Note:** Some UNIX systems use `ps -aux`, others `ps -ef` to check processes. Use the command appropriate for your operating system.

4. If there are old processes present that may be using a license, kill them.
5. If in step 2 you killed the license server, start it with the `runLm` command.

#### Preventing Multiple License Manager Processes

We recommend that you keep the license server running and that you not start and stop it as you run applications. If you prefer to stop the license server when you are not using it, always use `lmdown` to stop it.

Whenever you start the license manager, first run `lmstat` to be sure that there is not already a license server process running. Following this practice will ensure that you do not start multiple license servers.



# Index

---

## A

- address
  - MÄK Technologies vii
- applications
  - running 1-5

## C

- compatibility
  - network 1-3
- compiling 1-2, 1-4
- configuring
  - system for MÄK RT 2-2
- conventions
  - manual viii

## D

- debugging 1-7
- Dial-a-Sim vi
- directory
  - flexlm* 2-8
- DMSO
  - documentation 1-4
- documentation 1-4
- dynamic linking 1-3

## E

- environment variable
  - license manager 2-6
  - LM\_LICENSE\_FILE 2-6
  - RTI\_CONFIG 2-2
- environment variables 1-5, 2-2

## F

- fax
  - MÄK Technologies vii
- FED file 1-3, 2-2, 2-5
- file
  - descriptor 1-6
  - FED 1-3, 2-2, 2-5
  - license.dat* 2-6, 2-8
  - MTL 1-3
  - RID 1-3, 2-2, 2-4
- flags
  - linking 1-4
- flexlm* directory 2-8
- FLEXlm license manager 2-6

## I

- installing
  - MÄK RTI 2-2

## L

- library
  - libRTI.a 1-3
  - libRTI.so 1-3
  - rti.dll 1-3
- libRTI.a library 1-3
- libRTI.so library 1-3
- license manager 2-6
  - setting up 2-6
- license server 2-6
  - running 2-9
  - status 2-9
  - stopping 2-9

license, environment variable 2-6

*license.dat* file 2-6, 2-8

linking 1-4

dynamic 1-3

static 1-3

LM\_LICENSE\_FILE environment variable 2-6

*lmdown* program 2-9

*lmhostid* program 2-6, 2-8

*lmstat* program 2-9

Logger vi

## M

maintenance agreement vii

MÄK Logger vi

MÄK Plan View Display vi

MÄK RTI

configuring 2-2

installing 2-2

MÄK Stealth vi

MÄK Technical Support vii

MAK Technologies Lisp file 1-3

MÄK VR-Link Network Toolkit vi

manual

conventions viii

MOM data 1-6

MTL 1-3

## N

network compatibility 1-3

## P

packet server 1-7

parameters

RID file 2-4

phone number

MÄK Technologies vii

Plan View Display vi

products

Dial-a-Sim vi

Logger vi

Plan View Display vi

Stealth vi

technical support vii

upgrades vii

VR-Link vi

## R

recompiling 1-2

RID file 1-3, 2-2, 2-4

parameters 2-4

RTI

installing MÄK 2-2

services 1-6

RTI\_CONFIG environment variable 2-2

rti.dll library 1-3

rtidump utility 1-7

rtiexec 2-3

*runLm* program 2-9

running

applications 1-5

license server 2-9

## S

server 2-3

license 2-6

packet 1-7

services

RTI 1-6

setting up

license manager 2-6

shutting down

license server 2-9

Solaris

link flags 1-4

static linking 1-3

status

license server 2-9

Stealth vi

stopping

license server 2-9

support, technical vii

## T

technical support vii

## U

upgrades vii

User's Guide

organization of v

utility

rtidump 1-7

## **V**

variable

    license environment 2-6

variables

    environment 1-5, 2-2

VR-Link vi







**MÄK Technologies**  
*185 Alewife Brook Parkway*  
*Cambridge, MA 02138*  
*617.876.8085*