Intel NetStructure®
Host Media Processing

Diagnostics Guide

August 2005

05-2356-004

Publication Date: August 2005

Document Number: 05-2356-004

Intel Converged Communications, Inc.
1515 Route 10
Parsippany, NJ 07054

For **Technical Support**, visit the Intel Telecom Support Resources website at:
http://developer.intel.com/design/telecom/support

For **Products and Services Information**, visit the Intel Telecom and Compute Products website at:
http://www.intel.com/design/network/products/telecom

For **Sales Offices** and other contact information, visit the Buy Telecom Products page at:
http://www.intel.com/buy/networking/telecom.htm

Contents

Revision History 7

About This Publication 9

1 Diagnostics Overview 11

2 Tracing the Runtime Libraries 13

3 Getver Reference 15

4 KernelVer Reference 17

4.1 Description 17
4.2 Options 17

5 Runtime Trace Facility (RTF) Reference 19

5.1 Description 19
5.2 RTF Tool Architecture 20
5.2.1 Overview 20
5.2.2 Files Used by the RTF Tool 20
5.3 Starting and Stopping the RTF Tool 20
5.4 RTF Configuration File 21
5.4.1 RTFConfig Tag 22
5.4.2 Logfile Tag 24
5.4.3 Global Tag 25
5.4.4 GLabel Tag 25
5.4.5 GClient Tag 26
5.4.6 GClientLabel Tag 26
5.4.7 Module Tag 27
5.4.8 MLabel Tag 29
5.4.9 MClient Tag 30
5.4.10 MClientLabel Tag 30
5.4.11 Guidelines for Editing the RTF Configuration File 31
5.5 Example RTF Configuration Files 32
5.5.1 Example 1: Tracing disabled - RtfConfigWin.xml 32
5.5.2 Example 2: Tracing enabled, logfile path and size specified, one module configured - RTFConfigLinux.xml 32
5.5.3 Example 3: Tracing enabled, logfile path and size specified, several modules configured, global configuration used - RTFConfigWin.xml 33

Index 37

Figures

1 RTF Configuration File Tag Structure 22

Tables

1 RTF Configuration File Traceable Modules 27

Revision History

This revision history summarizes the changes made in each published version of this document.

Document No.	Publication Date	Description of Revisions
05-2356-004	August 2005	Global: Editorial and branding changes.
05-2356-003	April 2005	RTFConfig Tag: The default setting of the trace attribute has changed from 0 to 1. Example RTF Configuration Files: Added two Windows examples.
05-2356-002	March 2005	Diagnostics Overview: References to DM3StdErr and Qerror removed. Added paragraph and figure to explain how HMP products are shown on the DCM configuration manager GUI. Tracing the Runtime Libraries: Title of this chapter changed. Replaced the path of the location for the RTF tool with an environment variable. DM3StdErr Reference: Chapter removed. Getver Reference: Changed the example command and added a Note. Qerror Reference: Chapter removed. StrmStat Reference: Chapter removed. Runtime Trace Facility (RTF) Reference: Added a Note to the size topic in the Logfile Tag subsection. Guidelines for Editing the RTF Configuration File: A guideline was added about potentially experiencing I/O throughput degradation when using full RTF logging on high-density systems.
05-2356-001	September 2004	Initial version of document. Some of the information contained in this document was previously published in the *Intel® Dialogic® System Release 6.0 for CompactPCI on Linux Diagnostics Guide*, document number 05-1885-001 and the *Intel® Dialogic® System Release 6.0 for CompactPCI on Windows 2000 Diagnostics Guide*, document number 05-1935-001.

About This Publication

This preface provides the following information about this publication:

- Purpose
- Intended Audience
- How to Use This Publication
- Related Information

Purpose

This publication describes diagnostic tools, each of which is described in a separate reference chapter.

Intended Audience

This publication is intended for:

- System Integrators
- Toolkit Developers
- Independent Software Vendors (ISVs)
- Original Equipment Manufacturers (OEMs)

How to Use This Publication

To use this publication, look up the diagnostic tool you want to use in the Contents or Chapter 1, "Diagnostics Overview" or below and click the link to go to that chapter:

- Chapter 2, "Tracing the Runtime Libraries"
- Chapter 3, "Getver Reference"
- Chapter 4, "KernelVer Reference"
- Chapter 5, "Runtime Trace Facility (RTF) Reference"

This document also contains a preface (this chapter) and an Index.

Related Information

The following documents and web sites provide more information:

- *Intel NetStructure® Host Media Processing Software Installation Guide* - This document describes how to install the HMP Software.
- *Intel NetStructure® Host Media Processing Software Administration Guide* - This publication describes how to perform the various tasks related to obtaining, activating, and otherwise working with HMP Software license files and also how to manually stop and start HMP services.
- *Intel NetStructure® Host Media Processing Software Release Guide* - This document provides information about the release such as product features, system requirements, and user documentation.
- *Intel NetStructure® Host Media Processing Software Release Update* - This document addresses release issues such as known problems and documentation updates.
- For technical support, go to *http://developer.intel.com/design/telecom/support/*.
- For information about Intel NetStructure Host Media Processing products, go to *http://www.intel.com/design/network/products/telecom/software/index.htm#hmp*
- For information about Intel® telecom products, go to *http://www.intel.com/design/network/products/telecom/index.htm*.

Diagnostics Overview 1

This section provides a brief description of the diagnostic tools included with the system software. Each tool is described in a reference chapter.

Throughout this document, "HMP software resources" can be considered synonymous with "boards."

Getver Reference
: The Getver tool displays the version of any DM3 board binary file.

KernelVer Reference
: The KernelVer tool queries the board's kernel running on a particular processor for its version number.

Runtime Trace Facility (RTF) Reference
: The RTF tool provides a mechanism for tracing the execution path of various HMP runtime libraries. The trace information can be captured in log files or sent to a debug stream.

Tracing the Runtime Libraries 2

This chapter describes how to use the Runtime Trace Facility (RTF) tool for tracing the execution path of the Intel NetStructure® Host Media Processing (HMP) Software runtime libraries. For more details about the tool, including the tool's architecture, information about editing the RTF configuration file, and information about starting, stopping and restarting the tool's tracing capabilities, refer to Chapter 5, "Runtime Trace Facility (RTF) Reference".

Use the following procedure to customize and activate the RTF tool's tracing capabilities:

1. Locate the RTF configuration file. The default installation location is as follows:
   - `$(INTEL_DIALOGIC_DIR)\log` for Linux systems
   - `%INTEL_DIALOGIC_CFG%`[1] for Windows systems
2. If you are familiar with XML syntax, use any ASCII text editor to open the RTF configuration file (*RtfConfigWin.xml* for Windows and *RtfConfigLinux.xml* for Linux). If you are not familiar with XML syntax, open the RTF configuration file with XML editor software.
3. Edit the RTF configuration file to customize the RTF tool's tracing capabilities. Refer to Section 5.4, "RTF Configuration File", on page 21 for complete information about editing the RTF configuration file. The RTF configuration file includes the following XML tags:
   - **RTFConfig:** This tag's attributes configure the RTF tool itself. All other tags in the RTF configuration file are children of this tag. This tag must appear one time in the RTF configuration file. The RTF tool's tracing capabilities are turned off by default. At a minimum, you must change the RTFConfig tag's **trace** attribute setting from 0 (default) to 1 to activate the RTF tool. Refer to Section 5.4.1, "RTFConfig Tag", on page 22 for complete information about the RTFConfig tag. The RTFConfig tag includes the following child tags:
     - **Logfile:** This is the first child tag of the RTFConfig tag. The Logfile tag's attributes set the parameters for the RTF tool's logfile output. This tag is an optional part of the RTF configuration file. Refer to Section 5.4.2, "Logfile Tag", on page 24 for complete information about the Logfile tag.
     - **Global:** This is the second child tag of the RTFConfig tag. The Global tag is used to specify the global configuration. Configuration settings at the global level are valid for all modules in the RTF configuration file. When a module is traced at the global level, all activity related to that particular module is traced. Refer to Section 5.4.3, "Global Tag", on page 25 for complete information about the Global tag.
     - **Module**: This is the third child tag of the RTFConfig tag. The Module tag is used to specify the trace configuration for a particular module. The default RTF configuration contains pre-defined module tags for traceable runtime libraries. You can edit the **state** attributes of these pre-defined module tags or delete these pre-defined module tags. However, keep in mind that the *RtfConfigLinux.xml* file must contain at least one Module tag. Refer to Section 5.4.7, "Module Tag", on page 27 for complete information about the Module tag.

1. To find out what the INTEL_DIALOGIC_CFG directory is, type `echo %INTEL_DIALOGIC_CFG%` on a command prompt and note the path displayed.

4. When you are finished editing the RTF configuration file, save and close the file.
5. Start your application. As your application runs, the RTF tool will trace the runtime libraries according to RTF configuration file settings. Refer to the individual entries in the log file or debug stream to review the trace statements generated by the runtime libraries.
6. If you wish to dynamically edit the RTF tool trace levels while your application runs, it is not necessary to stop the RTF tool. Instead, perform the following:
   a. Open the RTF configuration file.
   b. Customize the settings in the RTF configuration file.
   c. Save and close the RTF configuration file.
   d. Issue the `RtfTrace -restart` command to restart the RTF tool.

   Note: Keep in mind that changes made to the RTF configuration file will not be reflected in the RTF tool output until the tool has been restarted.
7. At any time, you can issue the `RtfTrace -stop` command to stop the RTF tool. The RTF tool will not provide output while it is stopped.

Getver Reference 3

This chapter provides reference information about the Getver tool.

The Getver tool displays the version of any Intel NetStructure® DM3 architecture board binary file. It scans the binary for the standard DM3 version string and prints it to the screen.

Getver can also be used to get the version for most of the Linux binaries. The OA&M executable binaries and libraries have a version that getver can display.

The following example prints the version string of the *ssp.mlm* file to the screen:

```
getver ssp.mlm
```

Note: You must specify the path to the file if you do not execute Getver from the directory in which the file is located (in this case, the *data* directory).

KernelVer Reference 4

This chapter provides reference information about the kernelver tool. The following topics are provided:

- Description. 17
- Options. 17

4.1 Description

The KernelVer tool queries the board's kernel running on a particular processor for its version number. This tool can be used to verify whether or not a processor has crashed.

4.2 Options

The kernelver tool uses the following command line options:

-b<n>
: Logical ID of board (required).Use the listboards utility (Linux) or the Intel® Dialogic® Configuration Manager (Windows) to obtain the board's logical ID.

--d<level>
: Do not modify. Leave this at the default value of 0.

-f<file>
: Output file name (required to save output in a file).

-p<n>
: Processor number (required). Lowest allowable value is 1.

-l<n>
: Number of times the program will retrieve the version (optional). You can use this option to repeatedly ping the board's kernel to generate message traffic.

-h
: Displays the help screen.

-v
: Displays the version number.

The following example runs the KernelVer tool on board 1, processor 2:

```
kernelver -b1 -p2
```

Runtime Trace Facility (RTF) Reference 5

This chapter provides an overview of the Runtime Trace Facility (RTF) tool, including the tool architecture, a procedural overview for using the tool and information about editing the RTF configuration file (*RtfConfigLinux.xml* for Linux* and *RtfConfigWin.xml* for Windows*) file to set tracing configuration options. Reference information about starting, stopping and restarting the tool's tracing capabilities is also included. This chapter provides the following subsections:

- Description. 19
- RTF Tool Architecture. 20
- Starting and Stopping the RTF Tool 20
- RTF Configuration File 21
- Example RTF Configuration Files. 32

A procedure for using the tool is provided in Chapter 2, "Tracing the Runtime Libraries".

5.1 Description

The RTF tool provides a mechanism for tracing the execution path of the runtime libraries for the Intel NetStructure® Host Media Processing (HMP) software. The trace information can be captured in log files or sent to a debug stream. The resulting log file/debug stream output helps troubleshoot runtime issues for applications that are built with HMP software.

The RTF tool obtains trace control settings and output formatting from an RTF configuration file (*RtfConfigLinux.xml* for Linux and *RtfConfigWin.xml* for Windows). Each runtime library has several levels of tracing that can be dynamically enabled/disabled by editing the RTF configuration file while your application runs. This allows the RTF tool to be configured to meet specific needs.

Notes: ***1.*** If you run full RTF logging on high-density systems, you may experience I/O throughput degradation. It is recommended that you do not run RTF with full logging on high-density systems or any field-deployed systems. Instead, use just the default error-enabled logging.

2. When RTF logging is enabled, users should be aware that logs are stored in the \log directory (under *%INTEL_DIALOGIC_DIR%* for Windows or *$(INTEL_DIALOGIC_DIR)* for Linux) and take necessary precautions to ensure that the directory never fills.

5.2 RTF Tool Architecture

This section provides an architectural overview of the RTF tool. The following subsections are included:

- Overview
- Files Used by the RTF Tool

5.2.1 Overview

This subsection provides a brief overview of the RTF tool architecture. The top level of the RTF architecture consists of producers and consumers:

- A producer is a software component that has internal HMP software RTF APIs incorporated into its source code. This allows producers to generate trace information. The runtime libraries are examples of producers (e.g. Global Call, Device Management, Voice).
- A consumer is the internal RTF engine that receives trace information from the RTF producers, converts the trace information into a readable format and outputs the trace information to a log file or debug stream.

5.2.2 Files Used by the RTF Tool

The RTF tool uses the following files, all of which are installed as part of the HMP Software:

RtfConfigLinux.xml or *RtfConfigWin.xml*
: Editable file that allows you to customize the tracing and output capabilities of the RTF tool (e.g which runtime libraries the RTF tool will trace, the trace levels, location and size of log files etc.)

RtfConfig.dtd
: Document tag definition file for the RTF configuration file. This file is read only; *it should not be modified.*

Note: You must have administrative rights to modify the *RtfConfig*.xml* files.

5.3 Starting and Stopping the RTF Tool

The `RtfTrace` command is used to stop and start the RTF tool. This command is issued from the command line and relies on an environmental variable that is defined as part of the HMP software installation routine. Therefore, the `RftTrace` command can be issued from any directory.

The `RtfTrace` command has the following variations:

`RtfTrace -start`
: Initializes and starts the RTF tool. Tracing is only initiated if/when your application has been started.

`RtfTrace -stop`
: Stops the RTF tool.

```
RtfTrace -restart
```
Restarts the RTF tool.

5.4 RTF Configuration File

To make full use of the RTF tool, you will want to modify the RTF configuration file (*RtfConfigLinux.xml* for Linux, *RtfConfigWin.xml* for Windows). This configuration file allows you to define which trace messages will be included in the RTF tool output. This subsection provides some guidelines for modifying the RTF configuration file and reference information about the file's XML tags and attributes.

Notes: ***1.*** Keep in mind that the tags present in the XML file vary according to which HMP software release you are working with. For example, some tags in this section are only present in the *RtfConfigLinux.xml* file that ships with the HMP for Linux software while other tags are only present in the *RtfConfigWin.xml* that ships with the HMP for Windows software.

2. You must have administrative rights to modify the *RtfConfig*.xml* files.

The RTF configuration file uses a number of esoteric terms that should be understood before attempting to edit the file. The following definitions should be kept in mind as you edit the RTF configuration file:

module
: a binary file, typically an executable or a shared object library file (.so).

client
: an entity for identifying a device (e.g. "dxxxB1C1"), component (e.g. "WaveFileSource") or a function (e.g. "**dx_play**( )") that is to be traced by the RTF tool.

label
: an attribute associated with a trace statement (e.g. "Error", "Warning", "Info", "External API entry", "External API exit" etc.). A trace statement's label is used by the trace data output for categorization purposes.

trace entry
: individual entries in the trace data output. The trace data output is typically sent to a log file or debug stream.

0
: when a 0 appears next to a configuration item in the RTF configuration file, it indicates that the configuration item is disabled.

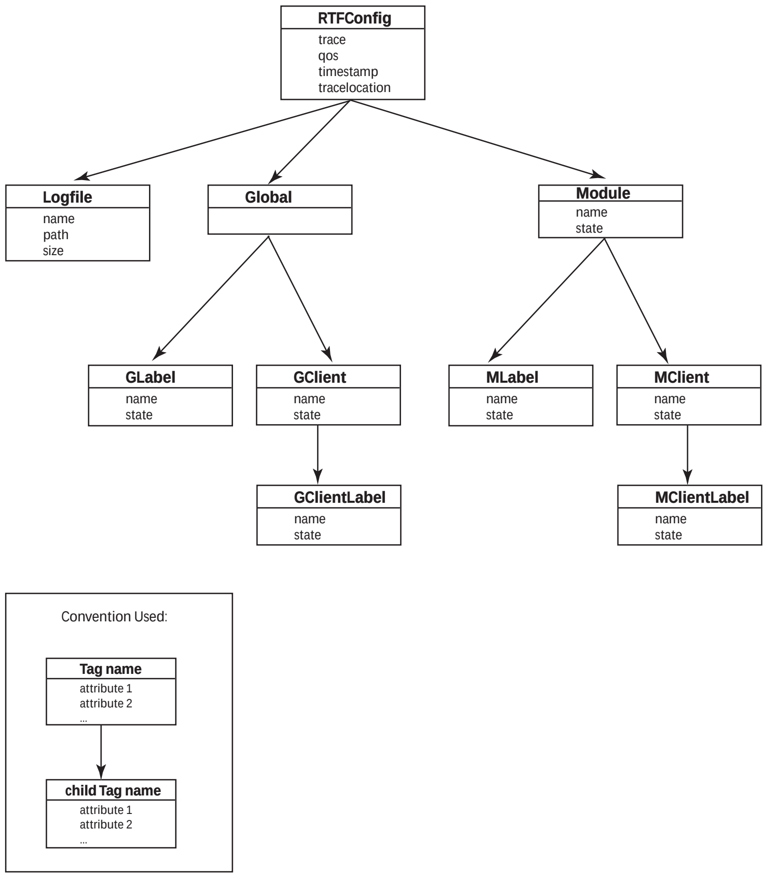
1
: when a 1 appears next to a configuration item in the RTF configuration file, it indicates that the configuration item is enabled.

The RTF configuration file's top-level document tag is the RTFConfig tag. The following three tags are child tags of the RTFConfig tag:

1. Logfile
2. Global
3. Module

Figure 1 shows the XML tag structure and tag attributes of the RTF configuration file:

Figure 1. RTF Configuration File Tag Structure



5.4.1 RTFConfig Tag

RTFConfig is the document tag. This tag is a mandatory component of the RTF configuration file; it can be found at the top of the RTF configuration file. A sample RTFConfig tag entry is shown below, keep in mind that certain tags are Windows-only while other tags are Linux-only:

```
<RTFConfig trace="1" tracelocation="TRACE_LOG" logformat="ALIGN">

<!-- Logfile section goes here -->

<!-- Global section goes here -->

<!-- Module sections go here -->

</RTFConfig>
```

The RTFConfig tag includes the following attributes:

trace

This attribute is used to enable or disable the RTF tracing capabilities. Valid values are as follows:

0
: RTF tracing is disabled.

1
: RTF tracing is enabled. This is the default setting.

tracelocation

This attribute determines the output mechanism for the RTF tool’s trace data. The trace data can be sent to a log file or a system-specific debug stream (e.g. **OutputDebugString( )** on Windows machines). Valid values are as follows:

TRACE_LOG
: RTF sends trace data to a log file. The log file details are specified in the Logfile tag. Refer to Section 5.4.2, “Logfile Tag”, on page 24 for more information. This value is the default.

SYSTEM_LOG
: RTF sends trace data to a system-specific debug stream (debug console on Windows and standard output console for Linux).

logformat

This attribute defines the format of the log file. The following two values are supported:

ALIGN
: Aligns the trace entry fields in the log file into comma separated columns. The top of each column includes a header that provides a description of the column’s content. This is the default setting.

UNALIGN
: Separates the trace entry fields in the log files with commas. Column alignment is not done.

Notes:
1. If you set the logformat attribute to ALIGN, you can customize the widths of the various columns. The following tags in the RTF configuration file allow you to define the aligned column widths:
   - **ModuleWidth number** - Allows you to customize the number of characters that appear in the Module column. The default setting is 10.
   - **ClientWidth number** - Allows you to customize the number of characters that appear in the Client column. The default setting is 15.
   - **LabelWidth number** - Allows you to customize the number of characters that appear in the Label column. The default setting is 10.
2. Using the ALIGN setting makes the log file easier to read but it does not make efficient use of hard drive space. This inefficiency is exacerbated as the log file grows. The UNALIGN format is separated by commas so it can be parsed by a spreadsheet or database program to make the file easier to read.

5.4.2 Logfile Tag

The Logfile tag is the first child tag of the RTFConfig tag. The Logfile tag provides logistical information about the log file(s) used to store trace output.

If you would like to customize the appearance and settings of the log file, edit the Logfile tag within the RTF configuration file. The Logfile tag appears under the RTFConfig tag in the RTF configuration file.

A sample Logfile tag entry from the *RtfConfigLinux.xml* file is shown below:

```
<Logfile path="$(INTEL_DIALOGIC_DIR)/log" size="1000" maxbackups="2"/>
```

A sample Logfile tag entry for the *RtfConfigWin.xml* file is shown below:

```
<Logfile path="$(INTEL_DIALOGIC_DIR)\log" size="1000" maxbackups="2"/>
```

The Logfile tag includes the following attributes:

path

Indicates a valid directory path for the log file. The default path for Linux is *$(INTEL_DIALOGIC_DIR)/log*. The default path for Windows is *%INTEL_DIALOGIC_DIR%\log*. The INTEL_DIALOGIC_DIR environment variable is defined as part of the HMP software installation routine.

For Windows systems, log files use an *rtflog-[date]-[time].txt* naming convention and cannot be changed. For Linux systems, the default log file name is *rtflog.txt*.

size

Sets the maximum size, in Kilobytes (KB), of the log file. The default setting is 1000. When the file reaches its maximum size, the RTF tool "rolls over" the log file, much like a circular buffer. RTF aligns the entries so that the oldest entry is always at the top of the log file and the newest entry is always at the bottom of the log file.

Note: Due to the internal buffers used by the RTF tool, the actual size of the log file may be up to 1 MB larger than the size attribute's value. For example, if the size attribute is set to 1000 KB, the actual log file may grow up to 2000 KB.

maxbackups

Indicates the maximum number of backup log files the RTF creates. If this attribute is set to 0, all trace information is written to one log file. If this attribute is set to 1 or greater, all trace information is initially written to one log file. When the size of this file reaches the threshold defined in the Logfile tag's size attribute, the RTF trace data "rolls over" into a second log file. This sequence occurs until the number of backup log files created equals the maxbackups attribute setting.The default value is 2.

5.4.3 Global Tag

The Global tag is the second child tag of the RTFConfig tag. The Global tag is used to specify the global configuration. Global configuration settings are valid for all modules included in the RTF configuration file. However, global configuration settings can be overridden by individual settings at the Module tag level (see Section 5.4.7, “Module Tag”, on page 27). The Global tag cannot occur more than one time in the RTF configuration file. The Global tag can either be empty:

```
<Global>

<!-- This is an example of an empty Global tag -->

</Global>
```

or the Global tag can have GLabel and/or GClient child tags. There are no attributes associated with the Global tag.

5.4.4 GLabel Tag

The GLabel tag is a child tag of the Global tag; it is used to configure global labels. If a label is defined at the GLabel level then all module and client behavior will be governed by this configuration (unless overridden for a given module at the MLabel level).

The following line may appear in the default *RtfConfigLinux.xml* file:

```
<GLabel name = “Error” state = “1”/>
```

This line indicates that tracing of all Error labels is turned on by default. To disable this default behavior, you can delete this line or change the “Error” state attribute setting to 0.

The following lines may appear in the default *RtfConfigWin.xml* file:

```
<GLabel name = “Error” state = “1”/>
<GLabel name = “Exception” state = “1”/>
```

These lines indicate that tracing of all Error labels and all Exception labels is turned on by default. To disable this default behavior, you can delete these lines or change the “Error” state attribute and “Exception” state attribute setting to 0.

The GLabel tag has the following two attributes:

name

Indicates the name of the global label to be configured. *You must define the name of the global label*; there is no default value. Possible labels include:

- “Error” (enabled at the Global level by default)
- “Debug”
- “Info”

- "Warning"

Note: Refer to the default RTF configuration file's `MLabel name` attributes. These default entries are for the HMP Software runtime libraries. Any of these runtime library label names can be included in a GLabel tag.

state

Specifies the state of the label. Valid values are as follows:

1
: Label is enabled at the global level. All trace messages associated with this label will be sent to the trace output. This is the default value.

0
: Label is disabled at the global level. Trace messages associated with this label will not be sent to the trace output.

5.4.5 GClient Tag

The GClient tag is a child tag of the Global tag; it is used to configure global clients (devices). If a client is defined at the GClient level then all client behavior will be governed by the this configuration (unless overridden for a given client at the MClient level). The GClient tag can be empty or have GClientLabel children tags. The GClient tag has the following two attributes:

name

Indicates the name of the client to be configured. *You must define the name of the global client*; there is no default value. Example clients include:

- "dxxxB1C1"
- "dxxxB2C2"

state

Specifies the state of the client. Valid values are as follows:

1
: Client is enabled at the global level. All trace messages associated with this client will be sent to the trace output. This is the default value.

0
: Client is disabled at the global level. Trace messages associated with this client will not be sent to the trace output.

5.4.6 GClientLabel Tag

The GClientLabel tag is a child tag of the GClient tag; it is used to specify a label for a global client. The GClientLabel tag has the following two attributes:

name

Indicates the name of a client label to be configured. *You must define the name of the client label*; there is no default value. Possible client labels include:

- "Error"
- "Warning"
- "Entry"

Note: Refer to the default RTF configuration file's `MLabel name` attributes. These default entries are for the HMP Software runtime libraries. Any of these runtime library label names can be included in a GClientLabel tag.

state

Specifies the state of the label. Valid values are as follows:

1
: Client label is enabled at the global level. Trace messages associated with this label will be sent to the trace output. This is the default value.

0
: client label is disabled at the global level. Trace messages associated with this label will not be sent to the trace output.

5.4.7 Module Tag

This tag is used to specify configuration for various modules. Configuration at the module level overrides global configuration. For example, if the state of a label "Error" is set to "1" in the global section and "Error" is set to "0" for an individual module, then the label "Error" will not be traced for that particular module. The module section must exist in the configuration file, even if the section is empty. Possible child tags of the Module tag are MClient and MLabel.

The RTF configuration file contains modules for the HMP Software runtime libraries. Table 1 lists the runtime libraries that have modules included in the default RTF configuration file. You can edit the state attributes of the modules to enable (set state = "1") or disable (set state = "0") the tracing. Alternatively, you can delete one or more modules from the default RTF configuration file if you are not interested in tracing certain runtime libraries.

Table 1. RTF Configuration File Traceable Modules

Library Description	File Name (.so for Linux, .dll for Windows)
PMAC Transport	pmac_transport
Fax library	libfax
Fax NTF library	libFaxntfmt (Windows only)
Voice library	libdxxm
Voice NTF library	voxspan (Windows only)

Table 1. RTF Configuration File Traceable Modules

Library Description	File Name (.so for Linux, .dll for Windows)
Standard Runtime library (SRL)	libsrl
NDI	ndi (Windows only)
IPM PMAC	libipm_pmac
Cheetah	Cheetah
BRI NTF	brintf (Windows only)
DTI NTF	libDtintfmt (Windows only)
DTI library	libdti
MSI Rev4 library	libmsir4 (Windows only)
MSI NTF	libMsintfmt (Windows only)
MSI DM3	MSI DM3 library
ODI	Dm3Odi
OTI	DM3Oti
DM3 Utility	DM3Utility
ipvsc library	libipm_ipvsc
ipm library	libipm
Global Call	libgc
Global Call (IP)	libgcipm
Global Call (PDK)	libpdkrt
Global Call Springware ISDN Translation layer	libgcis
ISDN library	libisdn
ISDN Technology Formatter library	isdnspan (Windows only)
UTI	UTI
CNF library	libcnf
Device management library	libdevmgmt
DM3 DCB	libDm3Dcb
IP CCLIB GC_H3R	gc_h3r
IP CCLIB SIP STACK	sip_stack
IP CCLIB H323 STACK	h323_stack
OAM	OAMSYSLOG

The Module tag includes the following attributes:

name

Indicates the name of a module to be configured. HMP Software runtime libraries have module names in the default RTF configuration file. Example module names in the RTF configuration file include:

- "cheetah"
- "libdcnf"
- "libdevmgmt"

state

Specifies the state of the module. Valid values are as follows:

1
: Module is enabled. Trace messages associated with this label will be sent to the trace output. This is the default value.

0
: Module is disabled. Trace messages associated with this label will not be sent to the trace output.

5.4.8 MLabel Tag

The MLabel tag is a child tag of the Module tag. The MLabel tag is used to configure module labels. If a label is defined at the global level and the same label is defined at the module level, the module level configuration overrides the global configuration for the module. The MLabel tag has the following two attributes:

name

Indicates the name of the label to be configured. The HMP Software runtime libraries have module label names in the default RTF configuration file. Example module label names in the RTF configuration file include:

- "Error"
- "Warning"
- "Entry"

state

Specifies the state of the label. Valid values are as follows:

1
: Label is enabled. Trace messages associated with this label will be sent to the trace output. This is the default value.

0

Label is disabled. Trace messages associated with this label will not be sent to the trace output.

Note: When the state attribute is not included or not defined, the default value is 1. However, the HMP Software runtime library module labels that exist in the default RTF configuration file have their state attribute initially set to 0.

5.4.9 MClient Tag

The MClient tag is a child tag of the Module tag; it is used to configure a specific client for the module. The MClient tag can be empty or have MClientLabel children tags. The MClient tag has the following two attributes:

name

Indicates the name of the client to be configured. The HMP Software runtime libraries have pre-defined module client names in the default RTF configuration file. Example clients include:

- "dxxxB1C1"
- "dxxxB2C2"

state

Specifies the state of the client. Valid values are as follows:

1

Client is enabled. Trace messages associated with this client will be sent to the trace output. This is the default value.

0

Client is disabled. Trace messages associated with this client will not be sent to the trace output.

Note: When the **state** attribute is not included or not defined, the default value is 1. However, the HMP Software runtime library module clients that exist in the default RTF configuration file have their state attribute initially set to 0.

5.4.10 MClientLabel Tag

The MClientLabel tag is a child tag of the MClient tag; it is used to specify a label for a client. The MClientLabel tag has the following two attributes

name

Indicates the name of a label to be configured. *You must define the name of the label*; there is no default value. Example labels include:

- "Error"
- "Warning"
- "Entry"

state

Specifies the state of the label. Valid values are as follows:

1
: Label is enabled. Trace messages associated with this label will be sent to the trace output. This is the default value.

0
: Label is disabled. Trace messages associated with this label will not be sent to the trace output

5.4.11 Guidelines for Editing the RTF Configuration File

Keep the following rules in mind when editing the RTF configuration file:

1. Do not change the name of the file. The filename must remain at its default setting.
2. The RTF configuration file is broken down into three sections. The sequence of the sections must always be as follows:
    a. Logfile configuration (not required)
    b. Global configuration
    c. Module configuration
3. The Logfile section, which provides logistical information about the resulting log file(s), is optional. If you do not provide information about the logfile, then the RTF tool will use the following log file settings:
    a. File name: *rtflog.txt* (default name is provided for Linux only)
    b. File location: *$(INTEL_DIALOGIC_DIR)/log* for Linux, *%INTEL_DIALOGIC_DIR%\log* for Windows
    c. Maximum size: 1000 KB
4. The Global configuration section can only appear one time in the RTF configuration file.
5. If a there is configuration information which conflicts in the global and module sections (for example, the global section enables a trace label for a client, but the module section disables this same label for the same client), then the module configuration overrides the global configuration.
6. Configuration information which is repeated later in the file will take precedence. For example, if there are two module configurations for *libdxxx.so* (Linux) or *libdxxx.dll* (Windows), the configuration information lower in the file will dictate tracing behavior for that module.

    Note: This should not be done, but it is mentioned here so the behavior can be recognized if this situation occurs by error.
7. RTF is case sensitive. Therefore, the trace labels “Error” and “error” are considered to be two distinctly different trace labels.
8. Simultaneous tracing of multiple HMP Software libraries may have an adverse effect on system performance.
9. If you run full RTF logging on high-density systems, you may experience I/O throughput degradation (specifically if you enable tracing on the MClient name MUTEX). It is recommended that you do not run RTF with full logging on high-density systems or any field-

deployed systems. Contact customer support before enabling extensive logging. Instead, use just the default error-enabled logging.

10. RTF affects running processes/applications only. If you enable RTF prior to starting your application, the RTF will not provide trace information until your application has started.
11. Save the original *RtfConfigWin.xml* file or *RtfConfigLinux.xml* file, as it is advisable to return to the original logging level when finished using the logging mechanism.

5.5 Example RTF Configuration Files

This section provides a number of example *RtfConfig*.xml* files along with a brief explanation of how the file settings effect the RTF tool's trace output. Note that the same rules/examples covered in this section apply to both the *RtfConfigWin.xml* and *RftConfigLinux.xml* file.

5.5.1 Example 1: Tracing disabled - RtfConfigWin.xml

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE RTFConfig SYSTEM "RTFConfig.dtd" >


<RTFConfig trace="0" tracelocation=”TRACE_LOG” logformat=”ALIGN”>


<Logfile path="$(INTEL_DIALOGIC_DIR)\log" size="1000" maxbackups=”2”/>

<Global>

</Global>


<Module name="libdxxmt.dll" state = "1">
    MLabel name="Error" state = "1"/>
    MLabel  name="Warning" state = "0"/>
</Module>


</RTFConfig >
```

Explanation

The RTFConfig tag's trace attribute is set to 0 so tracing is disabled. Trace output will not be created. Set the trace attribute to 1 to activate tracing.

5.5.2 Example 2: Tracing enabled, logfile path and size specified, one module configured - RTFConfigLinux.xml

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE RTFConfig SYSTEM "RTFConfig.dtd"  >
```

```
<RTFConfig trace="1">

<Logfile path="$(INTEL_DIALOGIC_DIR)/log" size="1024" maxbackups="2"/>

<Global>

</Global>

<Module name="libdxxx.so" state = "1">
   MLabel name="Error" state = "1"/>
   MLabel  name="Warning" state = "1"/>
</Module>

</RTFConfig >
```

Explanation

The RTFConfig tag's trace attribute is set to 1 so tracing is enabled. The Global tag is empty, so there is no global configuration. For this example, tracing is only configured at the module level.

The path for the log file is *$(INTEL_DIALOGIC_DIR)/log* and the maximum logfile size is 1024KB. The system maintains a maximum of 2 backup log files. The log file name is not specified so the default name (*rtflog.txt*) is used.

The only module configured for trace is *libdxxx.so*. This means that all clients (devices) for this module are configured to trace Error and Warning labels.

5.5.3 Example 3: Tracing enabled, logfile path and size specified, several modules configured, global configuration used - RTFConfigWin.xml

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE RTFConfig SYSTEM "RTFConfig.dtd" >
<RTFConfig trace="1" tracelocation="TRACE_LOG" logformat="ALIGN">

<Logfile path="$(INTEL_DIALOGIC_DIR)\log" size="1000" maxbackups="2"/>

<Global>
   <GLabel name="Entry" state = "1"/>
   <GClient name="dxxxB1C2" >
      <GClientLabel name="Exit" state ="1"/>
   </GClient>
</Global>



<Module name="libdxxmt.dll" state = "1">
   MLabel name="Error" state = "1"/>
   MLabel  name="Warning" state = "0"/>
</Module>
```

```
<Module name="libdxxmt.dll">
   <MLabel name="Entry" state = "0"/>
   <MLabel  name="Warning" state = "1"/>

   <MClient name="dxxxB1C2" >
      <MClientLabel name="Entry"/>
   /MClient>
</Module>


<Module name="libFaxntfmt.dll" state = "1">
   <MLabel  name="Warning" state = "1"/
</Module>



<Module name="libdtintfmt.dll">
   <MClient name="dxxxB2C1">
      <MClientLabel name="Internal_Exit"/
   </MClient>


</Module>

</RTFConfig >
```

Explanation

The trace attribute of the RTFConfig tag is set to 1 so tracing is enabled.

The logfile path is *$(INTEL_DIALOGIC_DIR)\log.* The log file's maximum size is 1000 KB. The system maintains a maximum of 2 backup log files.

The Global section is present in the configuration file, so all modules will have this global configuration. This configures "Entry" as a global label and "dxxxB1C2" is a global client for label "Exit".

In the module section there are two configurations for *libdxxmt.dll* so the later configuration will take precedence. Since the state of *libdxxmt.dll* is not specified it takes default value "1". Tracing is enabled for the module *libdxxmt.dll* but only for the "Warning" label. Even though "Entry" is configured to trace in the global section, it is disabled in the *libdxxmt.dll* module configuration, so the "Entry" label will not be traced.

The client "dxxxB1C2" is also configured to trace in the *libdxxmt.dll* module with the label "Entry". Therefore the client "dxxxB1C2" in *libdxxmt.dll* is traced for the label "Entry", even though it is disabled in the module section. This results in the client "dxxxB1C2" being traced for the labels "Exit" (from global configuration), "Warning" (from module configuration) and "Entry" (from its own configuration) in module *libdxxmt.dll*. While all other clients of *libdxxmt.dll* are configured to be traced for only label "Warning", as "Warning" is the only label configured to be traced for the module.

The *libFaxntfmt.dll* module is configured to trace “Warning” (from the module section) and “Entry” (from the global section). The *libFaxntfmt.dll* client “dxxxB1C2” is configured to trace for “Warning” (from module configuration), “Exit” (from global client section) and “Entry” (from global label section). All other clients of this module are configured to trace “Warning” (from module section) and “Entry” (from the global section) since these labels are configured for the module.

The *libdtintfmt.dll* module is configured to trace “Entry” (from the global section). “dxxxB1C2” in *libdtintfmt.dll* is configured to trace “Entry” (from the global section) and “Exit” (from the global client). The other client “dxxxB2C1” is configured to trace “Entry” (from the global configuration) and “Internalities” (from the module client section). All other clients of this module are configured to trace “Entry” (from the global section) since these labels are configured for the module.

Index

A

ALIGNED 23

B

binary file 11, 15

C

client 21
ClientWidth 23
consumer 20

D

document tag 22

E

Entry 27
Error 27

G

GClient tag 26
GClientLabel 26
Getver tool 11
getver tool 15
GLabel tag 25
Global tag 25

K

KernelVer tool 11, 17

L

label 21
LabelWidth 23
Logfile tag 24
logformat 23

M

MClient 27
MClient tag 30
MClientLabel tag 30
MLabel 27
MLabel tag 29
module 21
ModuleWidth 23

P

path 24
producer 20

R

RTF configuration file 21
RTF tool 13
RTFConfig tag 21
RtfConfig.dtd 20
RtfConfigLinux.xml 19, 20
RtfConfigWin.xml 19, 20
RtfTrace command 20
runtime libraries 13
Runtime Trace Facility 13

S

size 24
start the RTF tool 20
state 26
stop the RTF tool 20
SYSTEM_LOG 23

T

trace 23
trace entry 21
TRACE_LOG 23
tracelocation 23
Tracing disabled 32

U

UNALIGNED 23

V

version string 15

W

Warning 27