# Contents

## 1 Introduction
- 1.1 Idea and Core Features ........................................ 11
- 1.2 Standard Deployment Scenarios ............................... 11
- 1.3 Project Planning .................................................. 14
- 1.4 JMIF’s components ............................................... 16
  - 1.4.1 Link ......................................................... 16
  - 1.4.2 Gateway ..................................................... 16
  - 1.4.3 Channel .................................................... 17
  - 1.4.4 Dispatcher ............................................... 17
  - 1.4.5 Request and Response ................................. 17
  - 1.4.6 Telegram .................................................. 17
  - 1.4.7 Formatter ................................................ 17

## 2 Getting Started
- 2.1 System Requirements .......................................... 18
- 2.2 Installing JMIF .................................................. 18
- 2.3 Files and Folders ............................................... 19
- 2.4 Setting up JMIF ................................................ 20
- 2.5 Obtaining a License ........................................... 20
- 2.6 Starting JMIF .................................................. 22
- 2.7 Stopping JMIF .................................................. 23
- 2.8 Getting Support ................................................ 24
- 2.9 Uninstall JMIF - tidy thoroughly ......................... 24

## 3 JMIF Standard
- 3.1 Features .......................................................... 25
  - 3.1.1 Default Plug-Ins ...................................... 25
  - 3.1.2 Dispatcher ................................................ 25
  - 3.1.3 Channels .................................................. 26
  - 3.1.4 Devices ................................................... 27
  - 3.1.5 Work-flows ............................................... 29
  - 3.1.6 Error and Event Codes ............................. 31
- 3.2 Configuration .................................................. 32
  - 3.2.1 Naming Conventions ................................. 32
  - 3.2.2 Templates ............................................... 33
### Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.3 Telegrams</td>
<td>33</td>
</tr>
<tr>
<td>3.2.4 Regular Expressions</td>
<td>35</td>
</tr>
<tr>
<td>3.2.5 Display Formatting</td>
<td>36</td>
</tr>
<tr>
<td>4 JMIF as T3 Converter</td>
<td>38</td>
</tr>
<tr>
<td>4.1 Features</td>
<td>38</td>
</tr>
<tr>
<td>4.2 Configuration</td>
<td>39</td>
</tr>
<tr>
<td>5 Advanced Information</td>
<td>40</td>
</tr>
<tr>
<td>5.1 Creating a Scheduled Task</td>
<td>40</td>
</tr>
<tr>
<td>5.2 Running JMIF as Service or Daemon</td>
<td>40</td>
</tr>
<tr>
<td>5.2.1 Win32</td>
<td>40</td>
</tr>
<tr>
<td>5.2.2 Linux x86</td>
<td>41</td>
</tr>
<tr>
<td>5.2.3 Other Platforms</td>
<td>42</td>
</tr>
<tr>
<td>5.3 Backing up Installation</td>
<td>42</td>
</tr>
<tr>
<td>5.4 Migrating an Existing Installation</td>
<td>42</td>
</tr>
<tr>
<td>5.5 Quick Migration Steps</td>
<td>43</td>
</tr>
<tr>
<td>5.6 Major Upgrade on a Production System</td>
<td>43</td>
</tr>
<tr>
<td>5.7 Incompatibly Changed Settings</td>
<td>44</td>
</tr>
<tr>
<td>5.8 Controlling JMIFExternally</td>
<td>44</td>
</tr>
<tr>
<td>5.9 Shutdown Options</td>
<td>45</td>
</tr>
<tr>
<td>5.10 Using Chainsaw</td>
<td>45</td>
</tr>
<tr>
<td>5.11 Application Management and Monitoring</td>
<td>45</td>
</tr>
<tr>
<td>5.12 Changing System Properties</td>
<td>48</td>
</tr>
<tr>
<td>5.13 Using Includes for Property File based Configuration</td>
<td>48</td>
</tr>
<tr>
<td>5.14 Configure Loader</td>
<td>48</td>
</tr>
<tr>
<td>6 FAQ</td>
<td>50</td>
</tr>
<tr>
<td>6.1 Project Management</td>
<td>50</td>
</tr>
<tr>
<td>6.1.1 JMIF Standard vs Special</td>
<td>50</td>
</tr>
<tr>
<td>6.1.2 What’s the difference between JMIF and FastOrder?</td>
<td>50</td>
</tr>
<tr>
<td>6.1.3 Query Extended Storage Location Information</td>
<td>50</td>
</tr>
<tr>
<td>6.1.4 FTP Server</td>
<td>51</td>
</tr>
<tr>
<td>6.1.5 File vs. TCP Channel</td>
<td>51</td>
</tr>
<tr>
<td>6.1.6 What is JMIF’s telegram format</td>
<td>51</td>
</tr>
<tr>
<td>6.2 Security</td>
<td>51</td>
</tr>
<tr>
<td>6.2.1 Do I have to worry about used Java Runtime’s Security</td>
<td>51</td>
</tr>
</tbody>
</table>
6.2.2 Do I need administrative rights? ........................................... 52
6.2.3 Crypted values ................................................................. 52
6.2.4 TCP connections .............................................................. 52
6.2.5 Switch Port Security ......................................................... 52

6.3 Installation ........................................................................... 52
6.3.1 What Windows Versions are supported? .............................. 52
6.3.2 What is the recommended method on Windows? .................. 53
6.3.3 Can I install several JMIF instances on the same system? ... 53
6.3.4 Can I install several JMIF instances using different JRE version on the same system? ..................................................... 53
6.3.5 I’m concerned about Java’s performance ............................ 53
6.3.6 I’m concerned about Java’s security ................................. 53
6.3.7 How many storage devices could one JMIF handle? ............ 53
6.3.8 Does JMIF supports HoriOpt? ............................................. 53
6.3.9 Do I have to shutdown JMIF each day or week? .................. 53
6.3.10 POSIX: Continue JMIF on hangup signal .......................... 54
6.3.11 POSIX: JMIF start fails because configured Display could not be connected ......................................................... 54
6.3.12 Installing JMIF as a Service on Windows Vista or newer Windows Versions ............................................................... 54
6.3.13 Installing JMIF as a Service fails ...................................... 54
6.3.14 Starting JMIF as a Service fails ....................................... 54
6.3.15 How to solve The specified service has been marked for deletion Issue ............................................................ 54
6.3.16 WinNT 6.2 and 6.3 Tips and Tricks ................................. 55

6.4 Customization ..................................................................... 55
6.4.1 My key or value XYZ is not used or incorrect read ............. 55
6.4.2 Property Key Sequence or Grouping ................................. 55
6.4.3 My Windows path names does not work .......................... 55
6.4.4 My configured link does not work ................................. 55
6.4.5 The file filter does not match ....................................... 55
6.4.6 Working with the response header ID template and file names ................................................................. 55
6.4.7 Response messages undeliverable on TCP connections ........ 56
6.4.8 Number format problems on quantity ............................. 56
6.4.9 User input handling ....................................................... 56
6.4.10 Confirmation using external buttons ................................. 56
6.4.11 Response’s return code values should have other values .... 56
### Contents

6.5.10 Log Files Missing or Outdated .......................... 65
6.5.11 JMIF is idle ............................................ 65
6.5.12 Out Of Memory Exception ................................. 65
6.5.13 Updating JMIF during Runtime .......................... 65
6.5.14 Hide Console Window ................................... 65
6.5.15 Hide JMIF’s Loader Output ............................... 65
6.5.16 JMIF Service won’t stop ................................. 66
6.5.17 JMIF won’t stop ......................................... 66
6.5.18 JVM Crashed ............................................ 66

6.6 T3C .......................................................... 66
   6.6.1 T3C Mask State not updated ............................. 66

7 HOWTO ...................................................... 67
   7.1 Using Media Converter ..................................... 67
      7.1.1 Common FAQ ......................................... 67
      7.1.2 VSCOM .............................................. 68
      7.1.3 MOXA .............................................. 70
      7.1.4 Recommended Port Mode ............................... 72
      7.1.5 Summary ............................................ 73
   7.2 Connecting RLP Devices .................................... 73
      7.2.1 Basic Tests ......................................... 73
      7.2.2 Configuration and Calibrating ....................... 74
      7.2.3 RLP With Embedded LAN Interface ................... 74
   7.3 Connecting TIC Device .................................... 75
      7.3.1 Naming ............................................. 75
      7.3.2 Configuration ...................................... 75
      7.3.3 FAQ .............................................. 76
   7.4 C3000 Position Indicator Vario .......................... 77
   7.5 Dual Tray ............................................... 77
      7.5.1 Basics ............................................. 77
      7.5.2 How is it working? ................................. 77
   7.6 Modifying Support File Creation Behaviour .............. 80
   7.7 Firewalling ............................................. 80
   7.8 Building a JMIF Appliance ............................... 81

References ..................................................... 82
<table>
<thead>
<tr>
<th>Contents</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Legal Notice</td>
<td>83</td>
</tr>
<tr>
<td>History</td>
<td>84</td>
</tr>
</tbody>
</table>
List of Figures

1 Simple scenario showing some supported devices ................. 12
2 hostLINK scenario for SAP connection .......................... 13
3 JMIF as gateway between host and PowerPick .................... 14
4 JMIF Structure .............................................. 16
5 SFX Installer .................................................. 18
6 About Dialog: License found .................................... 22
7 About Dialog: License missing ................................. 22
8 Schedule Tasks List ........................................... 40
9 JMIF Service Properties ....................................... 41
10 Chainsaw GUI ................................................ 45
11 MBeans Browser (VisualVM) ................................... 46
12 MBean Property Manager (VisualVM) ......................... 47
13 C3000 Manager (VisualVM) .................................. 47
14 MBean RLP Manager (HTMLAdaptor) .......................... 48
15 DirectStore Timeouts ......................................... 57
16 TIC ............................................................. 76
17 Dual Tray ...................................................... 78
18 Dual Tray Sample ............................................. 78
19 Dual Tray Sample (cont.) .................................... 79
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interfaces and Dispatchers</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>Channel List</td>
<td>26</td>
</tr>
<tr>
<td>3</td>
<td>Device List</td>
<td>28</td>
</tr>
<tr>
<td>4</td>
<td>Work-flow Modes</td>
<td>29</td>
</tr>
<tr>
<td>5</td>
<td>Order Types</td>
<td>30</td>
</tr>
<tr>
<td>6</td>
<td>Mixed Mode Support</td>
<td>31</td>
</tr>
<tr>
<td>7</td>
<td>Codes</td>
<td>32</td>
</tr>
<tr>
<td>8</td>
<td>Incompatibly Changed Settings</td>
<td>44</td>
</tr>
<tr>
<td>9</td>
<td>JMIF-FastOrder Comparison</td>
<td>50</td>
</tr>
<tr>
<td>10</td>
<td>Storage Location Information</td>
<td>50</td>
</tr>
<tr>
<td>11</td>
<td>File vs. TCP Channel</td>
<td>51</td>
</tr>
<tr>
<td>12</td>
<td>Port Mode</td>
<td>72</td>
</tr>
<tr>
<td>13</td>
<td>Generic Serial Device Server Settings Matrix</td>
<td>73</td>
</tr>
<tr>
<td>14</td>
<td>Firewall Rules</td>
<td>80</td>
</tr>
<tr>
<td>15</td>
<td>Document History</td>
<td>87</td>
</tr>
</tbody>
</table>
Listings

1  Licensing ............................................................... 21
2  JMIF’s command line ................................................. 23
3  Sample nestat output ................................................. 64
4  NetCom - 413 Setup ................................................. 68
5  MOXA NPort-211 Setup ............................................... 70
1 Introduction

1.1 Idea and Core Features

The application Java Machine Interface (called: JMIF) is a common gateway. This gateway enables communication between two or more systems. Systems are storage units, conveyor technique, indicator lights, warehouse management systems (called: WMS) or enterprise resource planing software. While doing this JMIF offers the following possibilities:

- platform independence through Java
- based on established and experienced open source libraries
- runs locally without any system modifications
- Console-/ Service application (no GUI)
- plug-in enabled on all levels
- highly threaded
- running dozens of machines on one channel w/o any performance impacts
- extensive setting possibilities (partly at runtime) while using arbitrary multi host-store links
- format independent, centrally manageable settings
- extensive options for (centralized) monitoring

Note: The JMIF does not replace a warehouse management system. It works as a gateway between WMS and devices!

1.2 Standard Deployment Scenarios

The following deployment diagrams show various scenarios for running JMIF.
Figure 1: Simple scenario showing some supported devices

Figure 1 shows the most commonly used setup, where JMIF controls one or more devices triggered by a host system.
Figure 2 demonstrates a sophisticated SAP connection using our hostLINK module.

Figure 2: hostLINK scenario for SAP connection
The last sample figure 3 displays a deployment where JMIF operates as a gateway between a messaging middle-ware and our WMS named PowerPick. Some other scenarios:

- JMIF as T3Converter
- JMIF T3-TIC-Trap (controlling TIC display using given T3 telegrams)
- Driving with a bar code scan without connected host system

1.3 Project Planning

In order to obtain a successful JMIF project the following items should be clarified first:

- stakeholder list
- customer’s basic requirements
• deployment (host, devices, target platform)
• define request and response
• define work-flow
• define display format(s)
• specify host protocol
• specify telegram structure

Beside the standard there are already several defined interface (see table 1). Please contact KARDEX for detailed information especially if not standard dispatcher or special pick-by-light equipment is used.

<table>
<thead>
<tr>
<th>Vendor</th>
<th>System</th>
<th>Standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>aberle</td>
<td>Aberle-LVS</td>
<td>no</td>
</tr>
<tr>
<td>aberle</td>
<td>WACOS</td>
<td>yes</td>
</tr>
<tr>
<td>Allix</td>
<td>PROFFIX</td>
<td>yes</td>
</tr>
<tr>
<td>CIM</td>
<td>PROLAG</td>
<td>no</td>
</tr>
<tr>
<td>Consafe</td>
<td>Astro WMS</td>
<td>yes</td>
</tr>
<tr>
<td>Coscom/ Proxia</td>
<td>ToolDIRECTOR</td>
<td>yes</td>
</tr>
<tr>
<td>Jungheinrich</td>
<td>LI</td>
<td>yes</td>
</tr>
<tr>
<td>IGZ</td>
<td>SAP EWM/TRM</td>
<td>yes</td>
</tr>
<tr>
<td>inconso</td>
<td>SAP EWM</td>
<td>yes</td>
</tr>
<tr>
<td>KASTO</td>
<td>KASTOlvS</td>
<td>yes</td>
</tr>
<tr>
<td>KNAPP</td>
<td>KNAPP LVs</td>
<td>no</td>
</tr>
<tr>
<td>Klug-IS</td>
<td>iWACS</td>
<td>no</td>
</tr>
<tr>
<td>LS+/ Klinkhammer</td>
<td>DC21</td>
<td>no</td>
</tr>
<tr>
<td>LogControl</td>
<td>LogControl</td>
<td>yes</td>
</tr>
<tr>
<td>Moving</td>
<td>Moving WMS</td>
<td>yes</td>
</tr>
<tr>
<td>profil systems</td>
<td>profiLag</td>
<td>yes</td>
</tr>
<tr>
<td>proLogistik</td>
<td>pL-Store</td>
<td>yes</td>
</tr>
<tr>
<td>S&amp;P</td>
<td>SuPCIS</td>
<td>no</td>
</tr>
<tr>
<td>Sme.UP</td>
<td>iSeries Sme.UP</td>
<td>no</td>
</tr>
<tr>
<td>SHIL</td>
<td>(SPS)</td>
<td>no</td>
</tr>
<tr>
<td>Viastore</td>
<td>SAP EWM/TRM</td>
<td>yes</td>
</tr>
<tr>
<td>Zoller</td>
<td>TMS</td>
<td>yes</td>
</tr>
</tbody>
</table>

Table 1: Interfaces and Dispatchers

Connecting KARDEX Software like PowerPickGlobal, FastPic5, PowerPick5000 or
**1 Introduction**

*MegaLogistik*900 to ERP systems using message-oriented middleware (*MOM*) or raw TCP and file channel is possible, too. JMIF special packages are:

- RAWJMS: MOM connection (MQ Series or JMS)
- HL7GW: Health Level 7 data exchange
- HLFILE: SAP connection via MOM using hostLINK
- XMLHT: XML processing

**1.4 JMIF’s components**

Figure 4 shows JMIF’s basic structure used for configuration using components defined in the following sub sections.

![JMIF Structure](image)

**Figure 4: JMIF Structure**

**1.4.1 Link**

A *link* connects at least two systems. This is the start parameter to build up a communicating system. An arbitrary amount of links is definable. All configured links operate independently. A link has two gateways - one is called *Host* and the other *Machine*. Host and machine are just names to distinguish between both gateways and have only historical relationship to the real machines or host systems.

Furthermore a link manages the dispatchers and allows inter dispatcher communication.

**1.4.2 Gateway**

A *gateway* manages a list of channels and links the channels to one or more dispatchers.
1.4.3 Channel

A channel transports messages. Transport is based on send, receive and request. Channels may operate synchronously or asynchronously.

1.4.4 Dispatcher

A dispatcher manages and processes requests. Each request is stored as a thread (parallel processing) within a queue. The name of the queue defines the serialization criteria also called address or destination.

Note: JMIF special extensions are mainly contained in plug-ins. For that reason a dispatcher naming convention is given with the name of the host system.

1.4.5 Request and Response

Generally spoken a request is an order from the host gateway side of the link to JMIF. After the request is processed a response to the host side is created. Request or response represents in most scenarios an order position in the sense of a WMS.

1.4.6 Telegram

A telegram defines the request and response data structure as well as the data structures used on the machine side. Additionally the telegram manages the mapping to location coordinates such as address, tray, level, position or depth. A mapping decouples optionally host side coordinates from machine values (e.g. to control lamps).

1.4.7 Formatter

A formatter controls the layout of a display (normally within a panel on a physical machine).
2 Getting Started

2.1 System Requirements

The basic setup requires a Java Runtime Environment (called: JRE), Standard Edition SE version 1.6 or higher. Depending on the channel type there are also native drivers or components required (see section 2.2). A list of Java virtual machines could be obtained from Wikipedia.

Hardware requirements strongly depend on the selected operating system. We recommend using a system not older than three years in order to have JMIF run optimally. System should fulfill operating system vendor’s recommendations.

Virtualization is supported as long as the device’s port is reachable from the virtual machine.

Sufficient file system permission (full access for the user or service running JMIF) on JMIF’s application folder and on external transfer folders are required.

2.2 Installing JMIF

The required files are provided on CD, ftp, http (preferred), by email or via remote file transfer. Two installation types are currently supported. We highly recommend using a private JRE installation which is expected by the prepared scripts and found in the JRE sub-folder within JMIF’s application folder called jre/.

Methode 1 (SFX including JRE, Windows only): The installation is done by executing the given self extracting executable (called: SFX) and setting the target’s root folder.

Methode 2 (TGZ without JRE): The installation is done by decompressing the file jmif-vX.Y.Z.tar.gz into an arbitrary folder. This is done using a tool which supports GNU tar and GNU zip. Such a program should be available on any platform.
It’s possible to install and operate a JMIF version on one system several times using different folders. This applies also for different JREs and JMIF versions.

An installation script allows installation of platform specific drivers and places the required and delivered files into the JRE which runs the JMIF.

Note: If a system JRE is used, which is automatically updated, the install script must be started after each update. In this case merging of install and startjmif script is recommended.

2.3 Files and Folders

Important sub-folders to know are:

- **doc/**: contains the PDF manuals
- **jre/**: contains the private JRE
- **legal/**: contains legal stuff
- **lib/**: contains all deployed and required libraries
- **native/**: installation source of native and platform depended
- **tools/**: installation source of additional, external tools
- **transfer/**: commonly used file channel’s data exchange folder components

The following list describes the files which could be found after the installation in JMIF’s application folder:

- **jmif.jar**: the application itself
- **jmif.properties**: configuration file
- **stderr.log**: error messages which occurs before JMIF’s logger is loaded
- **jmif.log***: JMIF’s log file(s)
- ***.bat**: useful Windows batch files
- ***.sh**: useful Unix shell scripts
2.4 Setting up JMIF

Configuration of JMIF is done by editing JMIF’s property file. Only one plug-in supporting this kind of configuration storage is currently available. Plug-ins for databases or XML files can be provided on demand.

In most cases the property file is located in and used from JMIF’s application folder. The configuration file can be edited using any available text editor.

The file may contain key-value pairs, empty lines or comments. Sections such as Windows’ ini files are represented by text blocks containing keys with the same prefix (not necessary but recommended for readability).

A JMIF key starts with jmif. A hierarchical structure is achieved using dotted key notation.

A value can contain any kind of text except for

- backslash: creates a logical line break (must not be followed by any character in same line) otherwise it has to be doubled
- comma: separates values in a list, if value is considered as list of sub values
- dollar sign: if the right-hand value matches a key the value is resolved from that key (recursively)
- left- and right-hand blanks or tabs: removed during reading

System properties and environment variables are merged into the configuration. For example a JMIF key could refer to an environment variable’s value by using the dollar-syntax.

The property file is watched by the application itself. If the file is changed, JMIF disconnects, re-reads the configuration and connects. Therefore configuration changes can be done at runtime - including logger settings since v1.4.190 - which is useful for remote access or if JMIF running as service/daemon.

Furthermore JMIF’s configuration could be changed online using any available channel if it is configured as a configuration channel (see TechMan for more information). Please note that these changes are discarded on shutdown of JMIF.

2.5 Obtaining a License

In order to operate JMIF at least one license key is required. For clustering purposes it is also possible to use more than one key. The key(s) are available directly from KARDEX Software GmbH using the displayed registration information.
Note: The license code depends on all these values and furthermore on the fully qualified domain name (called: FQDN) of the system running JMIF. If the FQDN is not available JMIF tries the host name and finally the MAC address which depends on the undefined enumeration of all available network interface controllers (called: NIC). In the case of a support call the license information may be requested and checked by KARDEX Software GmbH!

Note: JMIF keeps running even if a license is missing! Therefore a valid license can be inserted during runtime without restarting the application.

Note: A demo mode is available but must be forced. This mode is not automatically activated if a license is missing! There are restrictions like operation delay, random shutdown/ limited runtime and required user interaction.

There are several ways to obtain a license:

- Phone: call KARDEX Software GmbH and provide at least order number and CodeId
- Mail: copy and paste the information above from the console output, the about dialog (see figures 6 and 7 or the log file into a plain text mail and send it out to KARDEX Software GmbH
- Mail with support file: create a support file and sent it out to KARDEX Software GmbH
- JMIF: start JMIF using command line option -m
2 Getting Started

![Image of About Dialog: License found](image1.png)

Figure 6: About Dialog: License found

![Image of About Dialog: License missing](image2.png)

Figure 7: About Dialog: License missing

2.6 Starting JMIF

JMIF supports following start methods:

1. Console: execute `java -jar jmif.jar`

2. Script: execute the startjmif script in JMIF’s application folder
3. Autostart: add the startjmif script to the user’s Autostart group

4. Scheduled Task/Init: add the startjmif script as scheduled task on system start (Windows, see section 5.1) or include the script call to the init script (Unix)

5. Service/Daemon: run JMIF as NT service or Unix daemon (see section 5.2)

Note: Because JMIF has its own loader the Java’s well-known CLASSPATH environment variable does not need to be set.

The Java application JMIF recognizes the following command line parameters:

<table>
<thead>
<tr>
<th>Usage: java -jar jmif.jar [OPTIONS]</th>
</tr>
</thead>
<tbody>
<tr>
<td>-S, --supportMail archive required files for support issues, send it per mail and exit</td>
</tr>
<tr>
<td>-c, --config &lt;file&gt; use given configuration file</td>
</tr>
<tr>
<td>-e, --encode &lt;value&gt; encode the given value</td>
</tr>
<tr>
<td>-h, --help print this message</td>
</tr>
<tr>
<td>-i, --install install operating system specific stuff and exit</td>
</tr>
<tr>
<td>-m, --mail mail license information to KARDEX Software GmbH and exit</td>
</tr>
<tr>
<td>-s, --support archive required files for support issues and exit</td>
</tr>
<tr>
<td>-t, --trayicon shows a tray icon</td>
</tr>
<tr>
<td>-u, --uninstall uninstall operating system specific stuff and exit</td>
</tr>
<tr>
<td>-v, --version print the version information and exit</td>
</tr>
</tbody>
</table>

Listing 2: JMIF’s command line

2.7 Stopping JMIF

JMIF supports the following stop methods which depending on the start method and the command line:

1. Console: press Ctrl-C

2. TrayIcon: select Shutdown or Hard Shutdown (see section 5.9)

3. Process: terminate the corresponding java process

4. Service/Daemon: stop the JMIF’s service/daemon using platform’s provided tool

5. Controller File: send HALT command (see section 5.8)
2.8 Getting Support

Should problems arise please check the FAQ from section 6 first. If JMIF still does not work call or mail your support contact providing JMIF’s configuration and log files. The easiest way is to send the created support request archive (see section 2.6 or use the support script) to your support contact. The support archive’s file name is made unique by using the current time-stamp.

*Note:* Support calls require ordered installation and configuration support or a valid support contract.

*Note:* If JMIF is started use the script named `ctrl_support` otherwise `support`.

2.9 Uninstall JMIF - tidy thoroughly

Uninstall platform depended service as well as native driver fragments using provided scripts and delete JMIF’s application folder.
3 JMIF Standard

3.1 Features

JMIF’s standard is defined by a default implementations for plug-ins, two dispatchers, channels and a list of supported devices.

3.1.1 Default Plug-ins

The following default plug-ins are available for the given components:

- Link: DefaultLink
- Gateway: DefaultGateway, DefaultTransformerGateway
- Formatter: DefaultDisplayFormatter

The DefaultGateway optionally supports persistence of messages and a simple transaction mechanism. Furthermore broadcasts and filtered broadcasts are supported.

The DefaultTransformerGateway extends the DefaultGateway with the capability to process all inbound and outbound messages by a transformer. Actually FilterMessageTransformer and XmlMessageTransformer are available as plug-ins.

3.1.2 Dispatcher

The standard package contains following dispatchers:

- DirectStoreDispatcher
- RlpDispatcher
### 3.1.3 Channels

<table>
<thead>
<tr>
<th>Type</th>
<th>Passive/Server</th>
<th>Active/Client</th>
<th>Streamed/Binary</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>extends File</td>
</tr>
<tr>
<td>FTP</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>items are whole files</td>
</tr>
<tr>
<td>Folder</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>SMTP, POP3, IMAP4 including SSL support</td>
</tr>
<tr>
<td>Mail</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>serial and parallel</td>
</tr>
<tr>
<td>JCA</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>1:1 connection</td>
</tr>
<tr>
<td>UDP</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>support n client connections</td>
</tr>
<tr>
<td>TCP</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>native WebSphere MQ</td>
</tr>
<tr>
<td>TCPServer</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>supports all messaging systems providing a JMS driver</td>
</tr>
<tr>
<td>MQS</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>actually AS/400’s MQ and IFS are supported</td>
</tr>
<tr>
<td>JMS</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>formerly named Jabber</td>
</tr>
<tr>
<td>ISeries</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>embedded http server</td>
</tr>
<tr>
<td>XMPP</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>dynamic WS invocation</td>
</tr>
<tr>
<td>HttpClient</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>WebSocket client</td>
</tr>
<tr>
<td>HttpServer</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>supports all DBMS providing a JDBC driver</td>
</tr>
<tr>
<td>DIWS</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>for piping purpose with other channels</td>
</tr>
<tr>
<td>WsClient</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>loopback</td>
</tr>
<tr>
<td>JDBC</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>SPS, currently only Win32 platforms are supported</td>
</tr>
<tr>
<td>XML</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>SPS, extends TCP</td>
</tr>
</tbody>
</table>

Table 2: Channel List

Channels may run synchronous (request) or asynchronous (send/ receive). Mixed mode is possible. Asynchronous receive is achieved by events, polling or scheduled jobs which
depend on the channel type. Most channels support configurable buffering on receive. Standard buffer implementation does not support prioritized messages at that layer.

Streamed channel could process data using following framing modes

- **NONE**: take all available data as message
- **CHAR**: end of message control character, optionally all except of STX, supports character stuffing
- **STRING**: end of message tag
- **PATTERN**: use pattern matching, optionally in combination with block length and fragment size
- **LEN**: fixed length
- **PACKET**: packet framing using combinations of fixed, dynamic, header and body (payload) length, optionally message type depended

*Note*: Availability of a channel type depends on the installed JMIF package. This applies to JMIF standard packages, too.

### 3.1.4 Devices

Table 3 shows a list of actually supported devices.
<table>
<thead>
<tr>
<th>Type</th>
<th>ControlUnit</th>
<th>Protocol</th>
<th>Vendor</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIMULATION</td>
<td>none</td>
<td>none</td>
<td>Kardex</td>
<td>Shuttle simulation</td>
</tr>
<tr>
<td>KX_C3K</td>
<td>C3000</td>
<td>C3000CGI</td>
<td>Kardex</td>
<td></td>
</tr>
<tr>
<td>KX</td>
<td>C2000</td>
<td>T3, T3TCP</td>
<td>Kardex</td>
<td></td>
</tr>
<tr>
<td>KX</td>
<td>LC3xx (C4000)</td>
<td>T3</td>
<td>Kardex</td>
<td></td>
</tr>
<tr>
<td>KX_DT</td>
<td>C2000</td>
<td>T3, T3TCP</td>
<td>Kardex</td>
<td>dual tray ready</td>
</tr>
<tr>
<td>KX_DT</td>
<td>LC3xx (C4000)</td>
<td>T3</td>
<td>Kardex</td>
<td>dual tray/ turn table ready</td>
</tr>
<tr>
<td>KX_RS</td>
<td>C2000</td>
<td>T3</td>
<td>Kardex</td>
<td>auto start mem</td>
</tr>
<tr>
<td>KX_AB</td>
<td>C2000, T88, T85</td>
<td>T3</td>
<td>Kardex</td>
<td>E1/E2</td>
</tr>
<tr>
<td>KX_AKHN</td>
<td>T88</td>
<td>T3</td>
<td>Kardex</td>
<td></td>
</tr>
<tr>
<td>KX_T4</td>
<td>C2000, T88, T85</td>
<td>T4</td>
<td>Kardex</td>
<td></td>
</tr>
<tr>
<td>RLP</td>
<td>RLP</td>
<td>RoyoTech</td>
<td></td>
<td>light pointer</td>
</tr>
<tr>
<td>EIM</td>
<td>EIM</td>
<td>TIC</td>
<td>Saunders</td>
<td>matrix display</td>
</tr>
<tr>
<td>MM</td>
<td>MM255/260</td>
<td>MM260</td>
<td>Megamat</td>
<td>dual tray ready</td>
</tr>
<tr>
<td>MM500</td>
<td>MM350/500</td>
<td>MM500</td>
<td>Megamat</td>
<td>drive operations</td>
</tr>
<tr>
<td>LISTA</td>
<td>LISTA</td>
<td>LISTA</td>
<td>Lista</td>
<td>PNS</td>
</tr>
<tr>
<td>INT</td>
<td>INT-MP</td>
<td>Net2</td>
<td>Intertex</td>
<td>Lift, Carousel, quantity corr.</td>
</tr>
<tr>
<td>INT</td>
<td>INT-EC1/INT-DSC1</td>
<td>Net2</td>
<td>Intertex</td>
<td>Carousel</td>
</tr>
<tr>
<td>INT</td>
<td>INT-DSC1</td>
<td>Net2</td>
<td>Intertex</td>
<td>Lift</td>
</tr>
<tr>
<td>SYS</td>
<td>MLT/MTH/IPN</td>
<td>SYST3</td>
<td>Schultheis</td>
<td></td>
</tr>
<tr>
<td>LM</td>
<td>LogiMat</td>
<td>OHLP</td>
<td>Handler</td>
<td>drive operations</td>
</tr>
<tr>
<td>LM20</td>
<td>LogiMat NewStyle</td>
<td>OHLP</td>
<td>Handler</td>
<td>drive operations</td>
</tr>
<tr>
<td>HA</td>
<td>MP0</td>
<td>MP0</td>
<td>Hänel</td>
<td></td>
</tr>
<tr>
<td>HA_MP10A</td>
<td>MP10A</td>
<td>MP10A</td>
<td>Hänel</td>
<td></td>
</tr>
<tr>
<td>HA_MP12D</td>
<td>MP12D</td>
<td>MP12D</td>
<td>Hänel</td>
<td></td>
</tr>
<tr>
<td>BTO</td>
<td>BTO</td>
<td>SMSP</td>
<td>Bertello</td>
<td></td>
</tr>
<tr>
<td>PD</td>
<td>PD</td>
<td>PD</td>
<td>Stanley</td>
<td>PowerDrawer</td>
</tr>
</tbody>
</table>

Table 3: Device List
Notes

- Device support may depend on installed JMIF standard package type.

- Foreign device support is limited due to lack of documentation, information and test equipment access. It depends on machine type, options and installed firmware version!

### 3.1.5 Work-flows

The DirectStore dispatcher supports following work-flows given in table 4.

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>default</td>
<td>fire’n forget, wait until arrived, wait until confirmed</td>
</tr>
<tr>
<td>1</td>
<td>shortest path</td>
<td>optimize by sorting, order based, supports internal dual tray</td>
</tr>
<tr>
<td>2</td>
<td>code based</td>
<td>code field defines E Memory</td>
</tr>
<tr>
<td>3</td>
<td>default cancel</td>
<td>cancel all requests on carrier equals 0</td>
</tr>
<tr>
<td>4</td>
<td>enhanced cancel</td>
<td>like 3 but creates cancel responses and allows confirmation by host (canceling of active request only)</td>
</tr>
<tr>
<td>5</td>
<td>cancel by new request</td>
<td>cancel active request if new one arrives (same as WFM 4 and request with code 50)</td>
</tr>
</tbody>
</table>

Table 4: Work-flow Modes

Work-flow mode 1 needs an end of order trigger. In case of file or TCP server channel the trigger could be created internally (end of file or connection closed). Otherwise the trigger has to be send from host system. In this case the boolean field *eoj* is mandatory.

In case of usage of *code* or *orderType* field, the work-flow could be controlled by host system, too. We call that order or booking type. Combinations with work-flow modes are possible. Difference between pick (out) and store (in) are usually used for different, order type dependent display formats.
<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>default</td>
<td>default mode, does not influence other settings</td>
</tr>
<tr>
<td>1</td>
<td>single in</td>
<td>stop processing on arrival</td>
</tr>
<tr>
<td>2</td>
<td>single out</td>
<td>stop processing on arrival</td>
</tr>
<tr>
<td>3</td>
<td>order in</td>
<td>force confirmation by operator</td>
</tr>
<tr>
<td>4</td>
<td>order out</td>
<td>force confirmation by operator</td>
</tr>
<tr>
<td>5</td>
<td>inventory</td>
<td>do not display quantity and force confirmation by operator with a valid quantity value</td>
</tr>
<tr>
<td>9</td>
<td>cancelation</td>
<td>cancel requests (depends on work-flow mode)</td>
</tr>
<tr>
<td>13</td>
<td>remove request</td>
<td>remove request from queue (only on work-flow mode 3 and based on hostID field)</td>
</tr>
<tr>
<td>40</td>
<td>compress</td>
<td>launch compression run (currently C3000 only)</td>
</tr>
<tr>
<td>41</td>
<td>add tray</td>
<td>add tray (currently C3000 only)</td>
</tr>
<tr>
<td>42</td>
<td>release tray</td>
<td>release tray (currently C3000 only)</td>
</tr>
<tr>
<td>50</td>
<td>acknowledge</td>
<td>confirmation by host (F3 simulation)</td>
</tr>
<tr>
<td>53</td>
<td>escape</td>
<td>escape by host (F4 simulation)</td>
</tr>
<tr>
<td>51</td>
<td>display</td>
<td>display text only, device depended!</td>
</tr>
<tr>
<td>52</td>
<td>status</td>
<td>query tray/opening/machine status only, device depended!</td>
</tr>
<tr>
<td>54</td>
<td>input</td>
<td>fetch user input from panel (currently C3000 only)</td>
</tr>
<tr>
<td>60</td>
<td>ignore</td>
<td>ignore request (usually used with mapped order types as filter)</td>
</tr>
<tr>
<td>61</td>
<td>alive</td>
<td>inter system alive request</td>
</tr>
<tr>
<td>62</td>
<td>queue state</td>
<td>query the queue state of opening (idle: code 0, busy: code 103, see table 7)</td>
</tr>
<tr>
<td>63</td>
<td>error description</td>
<td>query textual but configured return description</td>
</tr>
</tbody>
</table>

Table 5: Order Types
JMIF supports also toggling by operator between host mode (alias automatic mode) and manual mode (alias semi automatic mode), e.g. in case of rush order. This is called a *mixed mode workflow*.

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MM260</td>
<td>behaviour by device design turning off automatic start (default)</td>
</tr>
<tr>
<td>C2000</td>
<td>operator could re-request last tray inserting F2-F6-F2</td>
</tr>
<tr>
<td>C3000</td>
<td>mode detected by JMIF (fully automated)</td>
</tr>
</tbody>
</table>

Table 6: Mixed Mode Support

It’s recommend to adjust the corresponding timeouts (busy- and pollTimeout).

### 3.1.6 Error and Event Codes

Table 7 shows a list of JMIF’s standard error and event codes as well as machine error code ranges. Most commonly used code are 0, 101, 103 and 104.
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>no error, request successfully completed</td>
</tr>
<tr>
<td>1-99</td>
<td>T85/T88/C2000/MM260 machine error codes (short)</td>
</tr>
<tr>
<td></td>
<td>C3000CGI interface error codes (short)</td>
</tr>
<tr>
<td>101</td>
<td>common error</td>
</tr>
<tr>
<td>102</td>
<td>sequence number invalid</td>
</tr>
<tr>
<td>103</td>
<td>machine busy</td>
</tr>
<tr>
<td>104</td>
<td>timeout</td>
</tr>
<tr>
<td>105</td>
<td>maximum amount of retries reached</td>
</tr>
<tr>
<td>106</td>
<td>carrier in use or undefined</td>
</tr>
<tr>
<td>107</td>
<td>cancelled</td>
</tr>
<tr>
<td>108</td>
<td>invalid user input data</td>
</tr>
<tr>
<td>201</td>
<td>request accepted and queued</td>
</tr>
<tr>
<td>202</td>
<td>request processing started/ request active</td>
</tr>
<tr>
<td>203</td>
<td>carrier arrived, maybe overwritten by code 0</td>
</tr>
<tr>
<td>301</td>
<td>AO occupied with other tray on move back (Store/Put)</td>
</tr>
<tr>
<td>302</td>
<td>AO occupied with other tray on fetch (Pick)</td>
</tr>
<tr>
<td>501-999</td>
<td>MM260 FSE error codes</td>
</tr>
<tr>
<td>1000-32767</td>
<td>C2000TCP/C3000CGI machine error codes</td>
</tr>
<tr>
<td>0xFFFx</td>
<td>C3000CGI interface error codes (global, short)</td>
</tr>
<tr>
<td>&gt; 0xFFFF</td>
<td>C3000CGI interface error codes (long)</td>
</tr>
</tbody>
</table>

Table 7: Codes

3.2 Configuration

There are dozens of settings available. Check the TechManual, DirectStore Dispatcher for detailed information. Most of the settings could be defined globally in the sense of the dispatcher or locally in the sense of the target address.

3.2.1 Naming Conventions

Follow naming convention is recommended:

- Common: camelcase
- Link: name of the host system or generic host
3 JMIF Standard

- Gateway: compound of link name, host or machine and the word gateway
- Channel: uppercase letters, compound of channel type and number
- Host telegram: if request and response telegrams or identical just the word hostTelegram otherwise insert req or resp between the words host and telegram
- Machine telegram or command: compound of protocol, the word command and a number
- Display formatter: compound of the word displayFormatter and order type like (IN, OUT, INV)

3.2.2 Templates

JMIF uses simple internal templates. These templates are built up by using named fields (case sensitive!) surrounded with @@@ and any other static text. We call them @@@-template.

3.2.3 Telegrams

Telegrams on the host gateway are based on regular expressions (see [Friedl03]). The most important settings are:

- pattern: the regular expression used to match the given request data. The expression to parse or format a field value (depends on the fields data type).
- infoFieldNames: defines a list of additional fields. These fields can be used like the build-in fields.
- replaceTemplate: a regular expression or a @@@-template used to create the response data
- telegramSpecTemplate: can be used to define a sub telegram spec based on request’s data using @@@-template syntax (recursively usage possible)
- index: references the group within an regular expression. A group is defined by round brackets. The index is 1-based.
- country: the country argument is a valid ISO Country Code. These codes are upper-case, two-letter codes as defined by ISO-3166-1 (A2). You can find a full list of these codes at a number of sites, such as ISO-3166.
• **language**: the language argument is a valid ISO Language Code. These codes are lower-case, two-letter codes as defined by ISO-639-1 (A2). You can find a full list of these codes at a number of sites, such as Wikipedia List of ISO 639-1 codes.

• **defaultValue**: field’s initial value which has to fit to field’s data type specification.

Mapped fields can be referenced by adding the prefix `plain` before the field name, while uppercase the first letter of the name. The corresponding sub key could be created by placing a `@@`-template into the pattern key of the plain fields definition.

*Note:* Reverse mapping is not yet supported, i.e. machine data like a carrier number on a status request could not be reverse mapped.

Following variables are predefined. In most cases address and carrier are at least required. Unused but specified numeric fields have to be set to 0 whereas alphanumeric ones could contain an empty string:

• seqNo (integer): can be used to sequence number checks within used protocol, range configurable

• addr (integer, mapped): unique opening address

• sAddr (integer, mapped): unique source address for carrier transfer

• carrier (integer, mapped): machine wide unique carrier id

• carrierNext (integer, mapped): machine wide unique carrier id of carrier which should be prepared on lift (dual tray)

• level (integer, mapped): level on carousels

• pos (integer, mapped): position lamp (x axis, from)

• pos2 (integer, mapped): position lamp (x axis, to)

• depth (integer, mapped): depth lamp (y axis, from)

• depth2 (integer, mapped): depth lamp (y axis, to)

• hostId (string): can be used to join response and request on host side

• confirmFlag (boolean) : allows the host to ask for a response per request

• order (string): the order name or id

• orderType (integer, mapped): controls the workflow per request
• orderPos (string): the order position name or id
• part (string): the part name or id
• quant (double): the quantity, format pattern required
• desc (string): a description
• code (integer): combination of order type (request) and return value (response)
• ret (integer): return code inside response
• retDesc (string): textual description of the return code, freely configurable
• dateTime (date): timestamp

The machine gateway telegrams have machine type specific settings. Check the TechManual for details.

### 3.2.4 Regular Expressions

Most common used (sub-) expression within JMIF’s configuration are

- (): marks a group, group’s indices are 1-based
- (.*): matches any character, quantity: 0...n
- (\[^;\]+): matches any character except of semicolon, quantity: 1...n
- (\d{n}): matches any decimal character, quantity: n
- (\d){n,m}: matches any decimal character, quantity: n...m

Following characters have to be escaped by \\:

- dot: .
- hat: ^
- open curly brace: {
- open square bracket: [
- bracket: ( 
- pipe: |
• star: *
• plus: +
• questionmark: ?
• backslash: \\

More information about regular expressions (alias RegExp or just Regex) are available here:

• Books from O’Reilly: Mastering Regular Expressions [Friedl03], Regular Expressions Cookbook or Regular Expression Pocket Reference

• Wikipedia

• Regular-Expressions.info

• Regular Expression Test Page for Java RegexPlanet

• Online learn build and test tool RegExr

• Online regular expressions testing Rubular

• Online regular expressions testing regexe

3.2.5 Display Formatting

The default display formatting plug-in depends on the host gateway telegram, i.e. on the regular expression based telegram. Optionally the order type or code field could be used to create separate layouts depending on booking type. Formatting is carried out as follows, using:

• \textbf{R}: right padding

• \textbf{r}: get the rightmost characters

• \textbf{C}: center padding

• \textbf{c}: centers a string in a larger string using the space character

• \textbf{L}: left padding

• \textbf{l}: get the leftmost characters
• #: arbitrary character, where # is replaced by a space character while any other character is shown as filler

• .: decimal delimiter

• 0: zero or decimal, leading zeros, forced decimals places
4 JMF as T3 Converter

JMF could be used as T3 converter (called T3C). For this purpose an own package is available containing the needed plugins. JMF/T3C is able to convert the T3 protocol into the C3000CGI protocol.

**Warning** There could be several impacts which should be discussed with your system consultant! Such impacts are missing protocol features, timing issues, appliance on the panel, firmware versions or lack of communication resources.

**Note** T3C usage should be considered as workaround only. We strongly recommend using a common, machine protocol independent host interface like implementable with JMF standard. There is no guaranty that newer or next machine generations will be supported!

4.1 Features

JMF/T3C supports

- RS232 (including virtual COM ports like com0com), UDP and TCP
- all telegrams except of
  - R/S telegrams
  - Bahnhof functionality of x telegram
  - certain special functions of V telegram
- command and state caching
- command shelf time
- automatic start mask including E-Memory emulation
- T3 converter mask (several modes including AB buffer usage)
- host ID mapping
- trigger (configurable texts in P telegram)
- configurable error code mapping
- unlimited amount of machines per JMF instance
• up to six physical openings per machine
• T3Test v1.5 and MIF01 v3.7 proofed
• upto several thousand T3 requests per minute
• T3 trap (controlling RLP or TIC using given T3 data)

At this time there is no configuration wizard available. Furthermore JMIF has to be licensed. In case of T3 converter mask usage, the T3 converter option on machine side has to be licensed and activated, too.

4.2 Configuration

The T3C configuration requires at least the definition of the communication channel between host and JMIF, communication timeout, desired appliance on the panel, opening addressing and T3 telegram parametrization.

The configuration structure is JMIF standard except of the RLP dispatcher plugin has to be used (check the property file from the package as sample).
5 Advanced Information

5.1 Creating a Scheduled Task

Instead of using a service there is another way to achieve similar functionality on Windows platforms (running JMIF without a logged on user) - this is called scheduled task. To create a new task:

- open the Control Panel or Programs/Accessories/System Tools
- launch Scheduled Tasks
- double click Add Scheduled Task
- browse for startjmif.bat in Scheduled Task Wizards’s program selection
- name the task JMIF
- perform the task When my computer starts
- enter a valid local or domain account
- complete the wizard, you should have a list like in figure 8
- run the task

5.2 Running JMIF as Service or Daemon

5.2.1 Win32

The following steps are required to run JMIF as a Win32 service:

- copy all files from lib/servicewrapper/win32 to JMIF’s application folder
- modify the win32_wrapper.conf file on demand, e.g. set the service’s name, startup type and logging features
• execute the installservice script (or the uninstallservice script for uninstall purpose)
• adjust the log on settings (default: Local System, see figure 9)
• start the service once using service manager or just `net start JMIF` as administrator

5.2.2 Linux x86

The following steps are required to run JMIF as a daemon using init:

• copy all files from `lib/servicewrapper/linux-x86-32` to JMIF’s application folder
• modify the `jmif` and `wrapper.conf` files on demand, e.g. set the daemon’s name, paths, user and logging features
• execute `install jmif /etc/init.d`
• execute `update-rc.d jmif defaults`
• start the daemon once using `/etc/init.d/jmif start`

To remove JMIF daemon use:

• stop the daemon one using `/etc/init.d/jmif stop`
• execute `update-rc.d jmif remove`

• execute `rm /etc/init.d/jmif`

### 5.2.3 Other Platforms

At the time of writing this document only Win32 and Linux x86 binaries are deployed and supported.

*Note* The Win32 components works on a Windows based x64 system using the deployed JRE, too.

*Note* Because of licensing issues JMIF is made YAJSW (Yet Another Java Service Wrapper) ready. That means one could use YAJSW instead of the deployed framework.

### 5.3 Backing up Installation

Backing up an installation is simply done by

• stopping JMIF if started (see section 2.7)

• adding JMIF’s application folder recursively into an archive.

### 5.4 Migrating an Existing Installation

We recommend following approved plan in case of a scheduled migration to a new system:

• arrange a date with all stakeholders while considering downtime of the warehouse and availability requirements

• order the needed support from your KARDEX sales considering additional costs for nightwork, work at weekends and public holidays

• get and verify *direct* contact data of stakeholders

• check availability of stakeholders and systems a few days before the scheduled date

• ask for possible impacts

• prepare new communication data for reconfiguration, if they will be changed

• if possible install JMIF in advance to get license code

• consider all effected nodes in case of clustered systems

In case of a sudden system failure (ad hoc migration):
- failover cluster/SAN: node switch is done automatically, JMIF starts on new node and recovers transaction if activated
- cold stand-by system: boot-up system, JMIF starts if system is licensed before
- otherwise: restore from backup (see section 5.3) and obtain a new license code considering support level and related support times

Note At most cases the license will became invalid but JMIF still starts disconnected without a license.

Note Reconfiguration of media converters running in UDP mode will most likely be required in case of IP address of JMIF host was changed.

5.5 Quick Migration Steps

1. stop JMIF service on old system
2. copy JMIF’s application folder recursively to new system
3. disable JMIF on old system (e.g. by disabling the service or renaming the jmif.properties file)
4. install service on new system
5. start JMIF service
6. create a support file on new system and send it to KARDEX Software GmbH
7. insert new license code into jmif.properties and save the file
8. check JMIF’s log file for errors
9. remove JMIF from old system (delete service and then JMIF’s application folder)

5.6 Major Upgrade on a Production System

A major upgrade covers upgrading the private JRE, required libraries and JMIF itself. As very first step make a backup of the existing system as described in section 5.3.

Shutdown JMIF and rename the existing application folder. Install the new version using the old, original application folder name, copy the property file from the old installation and start JMIF. If necessary upgrade the license code. In most cases all other settings kept compatible and new ones use compatible default values.
In case of problems create a support file, shutdown JMIF and rename the current JMIF’s application folder. Rename the above saved folder to its original name and start JMIF again. You are back on your old productive system.

### 5.7 Incompatibly Changed Settings

<table>
<thead>
<tr>
<th>Component</th>
<th>Key</th>
<th>New</th>
<th>Since</th>
</tr>
</thead>
<tbody>
<tr>
<td>DirectStore</td>
<td>mmQuickStart</td>
<td>quickStart</td>
<td>1.4.106</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>renamed</td>
</tr>
<tr>
<td>DefaultDisplayFormatter</td>
<td>displayFormat</td>
<td></td>
<td>1.4.119</td>
</tr>
<tr>
<td></td>
<td></td>
<td>justification</td>
<td>mandatory</td>
</tr>
<tr>
<td>DirectStore/MM</td>
<td>driveMode</td>
<td></td>
<td>1.5.26</td>
</tr>
<tr>
<td></td>
<td></td>
<td>quickStart setting</td>
<td>mandatory on mode 61</td>
</tr>
<tr>
<td>TcpChannel</td>
<td>bufferSize</td>
<td>recvBufferSize</td>
<td>1.5.30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>renamed</td>
<td></td>
</tr>
<tr>
<td>DefaultGateway</td>
<td>eofOnRestart</td>
<td></td>
<td>1.6.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>removed due to violation of design</td>
<td></td>
</tr>
<tr>
<td>FileChannel</td>
<td>singleFileMode</td>
<td></td>
<td>1.6.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>changed to read only one file but do no longer also suspend channel (design violation)</td>
<td></td>
</tr>
<tr>
<td>DirectStore</td>
<td>resumeOnDone</td>
<td></td>
<td>1.6.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>removed due to violation of design use suspendUntilDone optionally in combination with WFM1 and/or singleFileMode</td>
<td></td>
</tr>
</tbody>
</table>

Table 8: Incompatibly Changed Settings

### 5.8 Controlling JMIF Externally

JMIF could be operated externally using the file `controller.jmif` which is located in the folder given by `user.dir`. This file is deleted on start and permanently watched during runtime. If the file is created or changed JMIF executes the given command. The following commands are currently valid:

- **CONNECT**: connect all links
- **DISCONNECT**: disconnect all links
- **RESTART**: disconnect and connect all links
- **HALT**: shutdown JMIF
- **SUPPORT**: create a support file
- **DUMP**: write thread dump to logger
5.9 Shutdown Options

JMIF has two shutdown options: one is the normal shutdown and the other is a hard shutdown. During a hard shutdown JMIF is immediately exited. In this case there is no disconnect, no shutdown of internal components and no open transactions will be finished. Within JMIF’s standard all transactions will be committed on a normal shutdown - it is immaterial whether they were successfully executed or just queued (purpose: system clean up). In order to continue working after a shutdown, having all unprocessed requests back, a hard shutdown using gateway persistence is required.

To force a hard shutdown if JMIF is running as a service or by the controller file’s command \textit{HALT}, the application setting \texttt{jmif.hardShutdown=true} has to be used.

5.10 Using Chainsaw

Monitoring JMIF’s activity could also be done by using the chainsaw script. This starts a graphical Java application called \textit{CHAINSAW Log4JLog Viewer} running a TCP server listing for incoming log entries from JMIF (see figure 10). To activate the logging towards Chainsaw the SocketAppender of Log4J (see samples in the given property files) must be setup.

Changing Chainsaw’s listening port (default: 4445) could be done by setting the system property \texttt{chainsaw.port}. Check the provided Chainsaw start script for a sample doing this.

As an alternative one could use \textit{Lilith} version 0.9.40 or newer.

5.11 Application Management and Monitoring

JMIF supports basic application management and monitoring including of connected devices. To activate JMIF’s Java Management Extensions (JMX) features, set the key \texttt{jmif.jmxSupport=true}. To enable simple i.e. insecure remote access use the following VM arguments:
5 Advanced Information

- `Dcom.sun.management.jmxremote`  
- `Dcom.sun.management.jmxremote.port = 9999`  
- `Dcom.sun.management.jmxremote.authenticate = false`  
- `Dcom.sun.management.jmxremote.ssl = false`.

Securing the remote access is strongly recommended (see JMX documentation). Web-based access is possible using JMIF’S extension `EmbeddedHtmlAdaptorServer` and the JMX tools.

Figure 11: MBeans Browser (VisualVM)
Advanced Information

Figure 12: MBean Property Manager (VisualVM)

Figure 13: C3000 Manager (VisualVM)
5.12 Changing System Properties

To overwrite or set system properties, the file `system.properties` located in JMIF’s application folder can be used. This may be required to change default settings of used libraries (e.g. internal mailing or XML handling).

5.13 Using Includes for Property File based Configuration

In case a property key starts with the word `include` JMIF will consider the property value as path for an other property file to include. Includes could be nested but recursion are checked.

5.14 Configure Loader

JMIF’s setup may not contain all components if specialized, commercial third party libraries are required.

Extending JMIF’s classloader is carried out using the file `jmifclassloader.properties` located in JMIF’s application folder. The syntax is defined by unique name equals path such as `FooBarComp-
nent=c:/components/foobar.jar. But one could also place such additional, required but not deployed runtime components in JMIF’s lib folder which is completely read by the loader. This is possible if the additional component was xcopy-deployed.
6 FAQ

6.1 Project Management

6.1.1 JMIF Standard vs Special

The common JMIF standard is represented by the DirectStore dispatcher. There are several other packages (plugin sets) available as standard in the sense of order management, too. Projects not covered by standard are implemented by plugins based on JMIF standard’s core and kept compatible with core also after the project was realized. In other words: special implementations are always integrated into main branch using plugins.

6.1.2 What’s the difference between JMIF and FastOrder?

<table>
<thead>
<tr>
<th>Feature</th>
<th>JMIF</th>
<th>FastOrder</th>
</tr>
</thead>
<tbody>
<tr>
<td>platform independent</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>database</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>GUI</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>WMS Setup</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>order sequence</td>
<td>by host</td>
<td>by user</td>
</tr>
<tr>
<td>order batching</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>order cancellation</td>
<td>by host/ by user</td>
<td>by user</td>
</tr>
<tr>
<td>pause and resume orders</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>process several orders in parallel</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>max. device count</td>
<td>no limits</td>
<td>4-8</td>
</tr>
</tbody>
</table>

Table 9: JMIF-FastOrder Comparison

6.1.3 Query Extended Storage Location Information

JMIF could provide internal storage location information if the connected device supports fetching such details.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Methods</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>C3000</td>
<td>request, JMX</td>
<td>height, weight, internal location coordinates</td>
</tr>
<tr>
<td></td>
<td>job, JMX</td>
<td>storage statistics and compression</td>
</tr>
<tr>
<td></td>
<td>job</td>
<td>sync machine’s tray tables to any external targets</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(e.g. a database)</td>
</tr>
</tbody>
</table>

Table 10: Storage Location Information
Note: The data may be inaccurate or may become invalid in case of manual storage action or storage compression!

6.1.4 FTP Server

The application of a FTP server could be combined with JMIF’s FileChannel. A FTP server is provided at most operating systems. Furthermore there are open source implementations available. Additionally JMIF has an extension (available on inquiry) which embeds a Java-based FTP server. This solution is platform independent, requires no installation nor system modifications and runs on a non-privileged port.

6.1.5 File vs. TCP Channel

Both channel types are widely spread to connect host systems. Using a simple CSV-based telegram specification and $(CR)(LF)$ as end of message tag they even could exchanged transparently on demand and at runtime.

<table>
<thead>
<tr>
<th>Channel</th>
<th>pros</th>
<th>cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>simple</td>
<td>performance</td>
</tr>
<tr>
<td></td>
<td>easy to test (notepad)</td>
<td>virus scanner</td>
</tr>
<tr>
<td>TCP</td>
<td>performance</td>
<td>department firewalls</td>
</tr>
<tr>
<td></td>
<td></td>
<td>testing requires a tool like telnet</td>
</tr>
</tbody>
</table>

Table 11: File vs. TCP Channel

6.1.6 What is JMIF’s telegram format

There is no one and only telegram format. JMIF could read information from

- any text string which could be parsed with an regular expression
- binary data
- markup languages using decomposer plugins.

Most commonly used is a simple character separated value message line containing an end of message tag for both directions. Data structure and transport are completely separated, too, i.e. one could exchange a channel while keeping the telegram’s structure.

6.2 Security

6.2.1 Do I have to worry about used Java Runtime’s Security

It’s recommended to use a private, xcopy-deployed JRE from a subfolder of JMIF’s application directory. In that case you don’t have to worry about Java runtime security issues because the runtime is not hooked to any Browsers or other applications. To prevent other users or processes from accessing that
6 FAQ

runtime, one could lock down JMIF’s application folder using file system permissions. Therefore there’s no need to update that runtime periodically.

A private JRE is nothing but a folder that contains JVM executables, setting files, required lib files, and extensions. You can install multiple JREs on your system. The JRE from JMIF is only known by JMIF’s launch scripts.

An JRE update of the used runtime after installation is not explicitly prohibited. Nevertheless in case of problems a rollback to the original JRE maybe required especially in case of a major JRE version number change. A rerun of the provided install-script after the JRE update may needed, too, for native components injected hereby into used JRE.

6.2.2 Do I need administrative rights?

No, as long as you do not need access to protected system resources (e.g. installing service/daemon). On Unix you could install and run JMIF from your home directory.

6.2.3 Crypted values

To encrypt a value use the command line option \( -e \). Clear the original key’s value, add a new key extended by the suffix \( -x \) and insert the encrypted value. Sample:

- original key: \( jmif.mySecretPassword = \)
- key containing encrypted value: \( jmif.mySecretPassword - x = 123456 \)

Any JMIF setting could be encrypted in this way. The values will not be displayed inside the log files. Encryption is runtime system dependent. Changing the runtime system will invalidate all encrypted values!

Alternatively it’s recommend to restrict access to JMIF’s application folder.

6.2.4 TCP connections

Like media converter JMIF’s TCP connections (passive and active) could be secured by configuring an expected password question and a valid password response. Don’t forget to append necessary control characters!

6.2.5 Switch Port Security

Note that some device may contain internal unmanaged switches. Machine subsystems beside controller unit may send data like broadcasts. Therefore all internal MAC addresses have to be configured to allow access and prevent port locks, i.e. add all to port’s ACL. Optionally you could connect each internal Ethernet device to an external switch with single secure configured MAC address per port.

6.3 Installation

6.3.1 What Windows Versions are supported?

Any version and architecture for which the minimal needed JRE is available. A x86 JRE works on x64 system types. A JMIF packages containing a x64 JRE is available on demand. In case of x64 platform ensure that all needed external native libraries are available as 64-bit version, too.
JRE 1.8 still working on Windows XP. Oracle has only dropped official support for Windows XP and
displays a warning about that fact during normal installation process, which is not used in case of JMIF
anyway (private JRE).

6.3.2 What is the recommended method on Windows?
Execute the delivered or downloaded SFX file and select a target folder without special characters or
blanks. Do not use your desktop!

6.3.3 Can I install several JMIF instances on the same system?
Yes (also as a service), but take care of the application’s lock setting and never ever use the same property
file.

6.3.4 Can I install several JMIF instances using different JRE version on the same
system?
Yes, there are no impacts or version conflicts as long as you consider the system requirements.

6.3.5 I’m concerned about Java’s performance
Forget it! By running JMIF on an old fashioned system it is possible to process several requests per
millisecond on the file channel. The device response times or poll intervals are factor 10 to 100 times
slower than JMIF’s internal routines.

6.3.6 I’m concerned about Java’s security
Simply use as recommended a private JRE. This JRE will never be used on a Web-Browser installed.
Furthermore the private JRE will not be touched by security updates or other (sometimes buggy) hotfixes
applied by auto updaters.

6.3.7 How many storage devices could one JMIF handle?
Well, that depends on a lot of factors like system performance, pick rate device type and request size.
Theoretical limits are give by JMIF’s internal data structures. But these are a far cry from any practical
application. JMIF runs productively in warehouses with dozens of devices without any performance
impacts. Until now JMIF has never been responsible for a system’s bottle neck.

6.3.8 Does JMIF supports HoriOpt?
Yes, this is a behaviour by design. JMIF drives machines in parallel. Also JMIF will control openings in
parallel on one machine if possible.

6.3.9 Do I have to shutdown JMIF each day or week?
No, JMIF is designed as service application running within a 24x7 environment. Sample log entries from
a mid range system running on Linux:
6.3.10 POSIX: Continue JMIF on hangup signal

To keep JMIF up and running even in case of a terminal session ends (e.g. by logout i.e. ignore the HUP signal) use:

```
nohup ./startjmif.sh </dev/null >/dev/null &
```

6.3.11 POSIX: JMIF start fails because configured Display could not be connected

In case JMIF should run without an UI simply disable the display using:

```
unset DISPLAY
```

Otherwise check your session settings so that configured X11 window server could be found.

6.3.12 Installing JMIF as a Service on Windows Vista or newer Windows Versions

Go to Start — All Programs — Accessories, then right-click on "Command Prompt", and select "Run as administrator". No UAC related issues are known.

6.3.13 Installing JMIF as a Service fails

Run the installer batch as administrator using JMIF’s application folder as working folder (console, JMIF application folder, command runas). If you prefer file explorer’s context menu, you have to adjust the JRE path inside the installer batch!

6.3.14 Starting JMIF as a Service fails

First of all check the wrapper’s log file and whether the configured JVM could be found and accessed. In case of a class not found exception check the PATH environment variable for invalid values. We saw double quoted entries on Windows systems there which caused corrupted values on java.library.path. If the variables value could not be fixed, remove the PATH reference from wrapper’s configuration file.

6.3.15 How to solve The specified service has been marked for deletion Issue

- A Microsoft Management Console (MMC) is opened. To ensure all instances are closed, run taskkill/F/IMmmc.exe. Note that services console and NT event viewer are hosted by MMC, too.
- SysInternals’ Process Explorer is opened. Closing it should lead to automatic removal of the service.
- Task Manager is opened.
6.3.16 WinNT 6.2 and 6.3 Tips and Tricks

For remote installation and assistance following notes may be helpful on Windows 8, 8.1, Server2012 and Server2012R2:

- starting a program: just search for it
- command shell: search for cmd or use Powershell
- logoff RDP session: run logoff inside command shell
- install telnet client: run pkgmgr /iu:"TelnetClient" as admin inside command shell

6.4 Customization

6.4.1 My key or value XYZ is not used or incorrect read

Configuration keys and values are case sensitive! Check the spelling! Check for copy'n paste errors! Check for doubled keys with different values! In case of logical line breaks check for invisible space characters after the backslash! Check that variable are resolvable! If JMIF miss a mandatory key, it writes the expected key name into the log file. Note that comments are not allowed inside a line. They will became part of the value! Further information regarding Java properties file format are available on Wikipedia.

6.4.2 Property Key Sequence or Grouping

Property keys are unsorted. It doesn’t matter were you place a key inside a property file. The grouping is just to keep similar things together and make the stuff more readable.

6.4.3 My Windows path names does not work

Check whether you doubled all backslashes. Use a forward-slash instead - it works! UNC path names and usage of administrative shares is supported as well.

6.4.4 My configured link does not work

Check the logs whether the link could be established. If not the links configurations or some of it's sub-components are wrong.

6.4.5 The file filter does not match

File filters base on regular expressions. Do not mistake that for file system, wild-card based expressions! For example a file system filter *.* correspond to JMIF’s filter expression (.*). Furthermore JMIF’s filter are case-sensitive. For example use (?i:.*\.txt) to fetch all files ending case-insensitive with .txt.

6.4.6 Working with the response header ID template and file names

The dispatcher’s template may contain all defined fields or constants. Furthermore it may contain path separators. But to use subfolders one should consider, that the folders have to exists in the up folder (only below version 1.4.200) and that there is no prefix defined. If the respHeaderIdTemplate setting is undefined, the request’s channel sender(From) information will be used for the response’s channel recipient (To).
6.4.7 Response messages undeliverable on TCP connections

Responses could not be sent if an active watch dog on a channel has closed the connection before the request could be finished. The watch dog timeout has to be adjusted or alive requests could be used. Another common issue is a defined but empty respHeaderIdTemplate. The setting should be disabled (removed or commented) except of use of TcpServerChannel with a maximum of one connection configured.

6.4.8 Number format problems on quantity

If one run into trouble with number format and different localisations using the quantity field, consider using an integer value or in case of informational purpose only any free text field.

6.4.9 User input handling

Except of inventory mode invalid input is accepted. JMIF will then use the values from request for response message. Commonly this is the requested quantity.

6.4.10 Confirmation using external buttons

- external button parallel to internal confirmation button: handled transparently
- T3: confirmation is mapped to input text using pattern #\{buttonnumber\}#. To use that information for response one could define a separate inputField. Hash tags could be remove using inputReplacements like #(\d)# → $1$.
- C3000: confirmation is mapped to OP’s input field and could be handled like with T3. Additionally one could use JMIF’s key code map feature.

6.4.11 Response’s return code values should have other values

Use the retDesc field to map the given return values to the required ones. In case of alive telegram use the code field to get value 0 or the ret field to receive value 61:

```
jmif.hostTelegram.replaceTemplate=@ret@@
jmif.hostTelegram.index.code=1
jmif.hostTelegram.index.ret=1
```

6.4.12 Prevent useless carrier move back operations

Commonly that’s a matter of the host system. In case move back requests could not be prevented, during request operation new request for the same carrier arrive and response is disabled or response sequence doesn’t matter, one could use following workarounds:

- prioritization: use code or order type as priority and enable PriorityQueues
- ignore: map code or order type of move back request to 60 and activate restoreOnDone
6.4.13 DirectStore’s Timeout Hell

There are three settings (see figure 15):

- busyTimeout: defines how long JMIF waits, until machine accepts drive command but only if machine clearly answers busy
- driveTimeout: defines how long JMIF waits, until requested tray moved to destination
- pollTimeout: defines how long JMIF waits, until operator confirmed on panel

6.4.14 RLP device will not work but initializes

Most likely one uses the quantity field in combination with rational numbers and internal notification (so called dispatcher observer’s). In this case the (internal) telegram has to be partly defined like this:

```plaintext
jmif.rlpDispatcher.internal.rlpRequest=rlpRequestTelegram
jmif.rlpRequestTelegram.pattern.quant==#,##0.000
jmif.rlpRequestTelegram.language.quant=DE
jmif.rlpRequestTelegram.country.quant=de
```

6.4.15 RLP device sporadically encounters an obstacle

Verify that
• tray dimension correctly setup (symptom: RLP initializes)
• sliding carriage operates without any barriers (symptom: X86/Y86 error)
• command write delay of 50msec inside UDP/TCP channel is configured (symptom: X86 error)

6.4.16 RLP deviation on Y-axis

Very probably one did not adapt yOffset and reference point to tray dimensions. Disable yOffset and pilot reference point (1098, cWidth/2, 0).

6.4.17 MM2xx Manual Memory Activated

If machine is not setup for direct start and poll mode two, active or in error state, new transport orders will be redirected from machine to its manual memory. These orders will not be processed until the operator pressed the start button on machine’s panel. To avoid this kind of problems, the workflow has to be defined carefully, the manual memory could be disabled by machine configuration and the poll mode two has to be activated.

In case the green button is still blinking and manual memory is disabled, then there’s an unconfirmed error on machine open which has to be resolved by operator. Furthermore following drive request will put into the manual memory but deleted immediately after operator has pressed the button and without any feedback to JMIF!

Note: Older controls like MM255 may not support disabling of manual memory! Older controls may not have poll mode two! In case of that and if a tray restore request was sent, JMIF is unable to detect completion of restore and releases therefore the opening immediately. Because machine may be still active on former restore, next transport request may be redirected by machine to manual memory.

6.4.18 MM2xx Tray Restore Freeze Machine or Panel

Most likely there’s a paternoster and not a lift in use. Disable usage of restore using subkey megalign = false.

6.4.19 MM255 Tray Restore Does Not Work

Some older firmware versions do not support that functionality as well as switching of position lamps. That’s no problem for JMIF’s request processing. But if that features are needed, please contact you KARDEX sales to request firmware special programming.

6.4.20 MM2xx Manual Memory Usage and Display

In case of
• a control which does not support extended drive command
• forbidden direct start
• requirement to use all four display lines
one could enforce this combination by
• disabling direct start on machine’s terminal
• using drive mode 61
• enabling deferred display but only if confirmation on terminal is required.

Starting with version 1.5.26 direct start (alias quick start) is no longer forced on drive mode 61.

6.4.21 Timeouts on C2000 and MM2xx

If communication channel working properly there’s most likely a mismatch between configured internal opening address inside machine and JMIF, i.e. JMIF uses an address, which is not known by machine. In case of

- MM2xx: check the configured address (7), Baudrate (8) and especially polling mode (9)
- C2000: check (service interface) whether MAC address is set (get Ethernet address)
- C2000: check (OP or service interface) whether Gateway is set (get ip gateway)
- C2000: check (service interface) configured protocol interface (get Ethernet protocol should return 0 for UDP and 1 for TCP)

6.4.22 C2000 NAK received

Check whether so called C2000’s host settings fit to JMIF’s T3 telegram settings. Default values:

- carrier: 3
- position: 2
- position2 alias bis..Position alias level: 0
- depth: 1
- quantity: 4
- display size: 20

6.4.23 T3 Timing Issues

Older devices (controls before C2000) as well as C2000 based Horizontals (HoCa) may have problems with to short polling intervals. Recommendation by factory is therefore to use a value not below 200msec (JMIF’s default is 350msec).

6.4.24 T3 Timeout Issues

Besides the above mentioned pitfalls on C2000 based controls consider following items on older controls supporting T3 protocol:

- opening address: double check opening address configured at machine and JMIF
- T85: enable t85 option at JMIF (symptom: no answer for status query command or machine control crash)
- display size: adjust display size inside JMIF to the configured one at machine, if unknown start with 20 (symptom: no answer for drive command)
- display missing: set display size inside JMIF to zero (symptom: even display size of 20 won’t work)
6.4.25 Unknown control and NAK received

Most likely you try to control a C3000 using a C2000 JMIF configuration.

6.4.26 Position Lamps Turned or Set Turned for Opposing Access Opening

- C3000: check parameter 207.x

6.4.27 C3000 Timeout Troubleshooting

There are two timeouts which should be configured to fit to each other. On the one side there are the HTTP channel timeout and retry values (communication layer). On the other side there is the command timeout (SPS layer). A recommended setup looks like this:

```
jmif.HTTP01.timeout = 1500
jmif.HTTP01.retries = 2
...
jmif.c3kCommand1.timeout = 3000
```

This means that on the communication layer one http request may timeout before the SPS command itself will fail. Finally use

```
jmif.HTTP01.forceRetries = true
```

to run retries on any I/O errors (not only http ones).

6.4.28 Disable host mode check

While this is not recommended one would like to turn off this feature (e.g. to use device’s option to switch automatically from user to host mode mask). Inside standard this could be done using the `checkHostMode` setting either globally or locally (i.e. per access opening). Please note, that in this case a tray directly fetched by the user may be restored immediately on a new host request! In case of trouble check the machine setup!

6.4.29 HTTP Proxy Information

At this time there are no problems known using JMIF, HTTP communication and proxies because JMIF does not use system’s HTTP communication components. To get some HTTP proxy information

- set the debug log level
- configure a channel of type `HttpClientChannel`
- specify an valid URL (otherwise the application vendor’s URL is used)

The proxy configuration is then printed to the log file while the channel is opened.
6.4.30 Verify C3000 Control Connectivity

To verify that a given IP address belongs to a C3000 control, open a TCP connection on port 81 using a Web Browser or telnet. In case the IP address belongs to an OP, connection will fail otherwise you should receive a status page or just a HTTP response.

6.4.31 C3000 acts strange and showing mysterious texts

Most likely T3 Converter option is active and machine showing T3 mask. Disable machine option carefully having access opening cleared or call Kardex service. After disabling T3 option you may run into some problems. In case machine reports tray already exists, you had a tray in the opening on deactivation of T3 option. Re-request the same tray again in semi-automatic mode and then restore it manually.

6.4.32 C3000 Measurement Units

Height of a tray in steps of 25mm. Minimum height equals to 3, i.e. 75mm. Weight of a tray’s payload in kg, i.e. net weight.

6.4.33 Intertex Controls seems to be slow

Check JMIF’s logs for time gaps of round about 4 seconds \((4s + (\text{networkAddress} \times 20\text{msec}))\). This is the interval within the control repeats unconfirmed messages. In case of media converter usage choose port mode TCP Server.

6.4.34 JDBC Connections

Using JRE’s JDBC-ODBC bridge in productive environments is not recommended. We suggest using Type 4 drivers (native protocol driver) which are platform independent and do not need installation of any additional client software besides the library file. The given library file could be placed in JMIF’s lib folder (xcopy) using an arbitrary filename and is usually freely available from DBMS vendor’s homepage.

For details of the database connection’s parameters contact your DBMS administrator.

- Microsoft SQL Server (Type 4 Driver)
  - Library: sqljdbc\4.0\sqljdbc.jar
  - Driver: com.microsoft.sqlserver.jdbc.SQLServerDriver
  - URL: jdbc:sqlserver://[serverName][\instanceName][; portNumber]]
    [:property = value][;property = value]]
  - Typical properties: databaseName, instanceName, integratedSecurity
  - Typical values: portNumber = 1433
  - Sample:
    jdbc:sqlserver://localhost\SQLEXPRESS:1433;
databaseName = JMIF;
  - Note: Connecting with Integrated Authentication works only on Windows systems. Use JMIF’s installation script to copy the required DLL to used JRE.
Note: Instance name could be omitted if default instance is used (but do not just clear attribute's value - remove key and value!).

Note: SQL scripts could be configured using semicolon as command separator.

- Oracle (Type 4 Driver alias Thin Driver)
  - Library: oracle12cr1 – ojdbc16.jar
  - Driver: oracle.jdbc.OracleDriver
  - URL:
    jdbc:oracle:
    drivertype : @[serverName] : [portNumber] : [SID] or
    drivertype : @[serverName] : [portNumber]/[serviceName]
  - Typical properties: n/a
  - Typical values: drivertype = thin, portNumber = 1521
  - Sample:
    jdbc:oracle:thin : @localhost : 1521 : orcl or
    jdbc:oracle:thin : @localhost : 1521/orcl
  - Note: SQL scripts could be configured using anonymous PL/SQL blocks.

6.4.35 Supported Encodings

Most channels support the charSet property. A list of supported encoding could be found inside the Java SE 8 Supported Encodings documentation.

In case of C3000 based machines, ensure that installed Language Pack, selected language, OPgraphic/OPtouch version (> 1.07.59), OPbrowser version (≥ 1.5.14.103) and JMIF’s http encoding match. For example for Russian (Cyrillic script) select Language Pack 2, language Russian and encoding cp1251.

6.4.36 Undefined destination error although settings are ok

Check on JMIF start or connect, whether the corresponding channel could be opened. Most likely there’s a resource conflict e.g. a configured port is used by another application on the system.

6.4.37 JMIF is unable to create a work file on upload

Some systems create a so called trigger file. If the file channel’s appending mode is switched on, the trigger file is empty and there’s no other error messages around, then you have to switch off the trigger file on host side.

6.5 Operation

6.5.1 License missing error occurs

Most probably you never have a valid license or the systems FQDN was changed. Also check that you have a valid amount of machine channels licensed and that the other license information did not change.
6.5.2 Why the hell JMIF runs idle without a license

There are several reasons:

- JMIF could be licensed and configured online by changing the property file, especially if JMIF runs as service or daemon on a clustered or cold stand-by system. Furthermore JMIF need to be started to get the license data. But a shutdown and restart after one has obtained the license is not wished.
- JMIF could be licensed and configured over a channel. This is used, if the configuring system is licensing JMIF on demand.
- There could be special JMIF plugins with separate licensing models.
- Demo Mode.

6.5.3 JMIF stops working after system shutdown

JMIF runs as service or scheduled task on Windows and the used NT-account’s password expired. Take the local system account if no network resources are used.

6.5.4 Could JMIF clutter the file-system with rubbish?

No. The log file appender uses log rotating with a certain amount of files and a maximum file size. The file channels history has a configurable cleanup worker which could also be used on the outbound folder.

6.5.5 JMIF hangs

Most likely it seems that JMIF hangs because of a far too big timeout value specified somewhere. Please note the unit of measure per setting! Other well-known possible reasons:

- Windows console: JMIF’s runs inside a console and someone has marked some text inside. This will lock JMIF, too!
- AntiVirus software update on Windows: may lock all other processes for seconds (we saw some updater blocking the system upto 40 seconds)
- huge system’s date/time adjustments: symptoms are lack of log entries over a certain amount of time (minutes) and negative logged measurement results. Please check your system logs!
- usage of asynchronous communication while channel’s listener mode is not enabled in combination with huge timeout values
- log file inaccessible or corrupted e.g. on SSDs: turn off logging into files by removing log file appender (log4j.rootLogger=debug, stdout) and set log file level on file logging to NONE (wrapper.logfile.loglevel=NONE)

In case you could exclude all of these reasons and JMIF continues after a restart, please execute the ctrl_dump batch before you restart it next time and before you execute the support file creation batch.
6.5.6 Starting JMIF fails

The most common reasons are:

- missing JRE: check the paths inside the start scripts
- another instance running: check whether JMIF is already started or if one need several instances
  check the appLock setting
- another application is using JMIF’s default port 52525 used for application locking: disable the
  application lock or configure another, unused port
- a local firewall prevents opening a listening TCP port: adjust firewall’s configuration
- native components not installed: run the install script
- all JMIF’s application folder’s subfolder are gone or never exists (package was not recursively
  extracted or copied)

6.5.7 Process ID of JMIF’s JVM

Starting with version 1.5.25 the PID is logged within the uptime information output.

If this fails (it depends on the JVM implementation) or an older version is used, try to search for an
configured port or the application lock port using a command like `netstat -abn`:

```
Proto Local Address Foreign Address State PID
... TCP 0.0.0.0:15008 0.0.0.0:0 LISTENING 4904 [ java.exe ]
TCP 0.0.0.0:52525 0.0.0.0:0 LISTENING 4904 [ java.exe ]
```

Listing 3: Sample netstat output

Alternatively you could use `jps` from JDK and look for a Java process called `JmifLoader` or `JmifServiceWrapper` if JMIF runs as service or daemon.

Finally you could look for the system property `wrapper.java.pid` in JMIF’s log files in case of JMIF
running as service or daemon.

6.5.8 Log File Analysis

Log files could be analyzed and watched live by using tools like `grep` and `less`. For Windows we suggest
e.g. the tools named `baregrep` and `baretail` which have graphical user interfaces. Do not use tools which
lock the files! In case of Unix-like (*nix) operating system prefer “`less +F`” over “`tail -f`”. The same is
true for watching files from FileChannel.

6.5.9 Reset Log File

For testing or issue analyses purpose a reset of the log file on JMIF start could be helpful. Use following
settings:

```
log4j.appender.rollingFile.MaxBackupIndex = 0
log4j.appender.rollingFile.append = false
```
6.5.10 Log Files Missing or Outdated

Log files are created by a so called file appender. Appenders do not have to work to run the application. Therefore JMIF may run without having log files or with outdated log file content. A file appender may not work, if the log file’s attribute is set to read only, log file’s permissions were changed (e.g. different user launching JMIF), log file path is unavailable, log file’s storage system run out of space or log file is locked by another application. Check file stderr.log or console output for error details.

6.5.11 JMIF is idle

This could have several reasons:

- license is missing: see section 2.5
- invalid configuration: check the log files for errors
- disconnected: JMIF was disconnected by tray icon, controller file or configuration channel
- no requests: all requests were processed successfully or with errors (check log files and JMIF’s responses)

6.5.12 Out Of Memory Exception

Please check the JMIF start script or the service wrappers configuration.

- start script: maximum heap size should be set to 128Mbyte using the option \(-Xmx128m\).
- service: check the win32_wrapper.conf file for key

\[wrapper.java.maxmemory = 128.\]

In case of ongoing trouble increase the value to 256Mbyte.

6.5.13 Updating JMIF during Runtime

If JMIF runs as a service, JMIF could be updated even without having access to a console or administrative rights. This requires following settings in the service wrapper’s configuration files as well as the HALT command:

- \[wrapper.restart.delay = 20\]
- \[wrapper.on.exit.default = RESTART\]

After the HALT command is sent one have 20 seconds to exchange the jar file.

6.5.14 Hide Console Window

Start JMIF using javaw instead of java. On Windows systems insert the start command in front of the command line. For other system use an ampersand after the command line.

6.5.15 Hide JMIF’s Loader Output

Because of that at this time no logger is available, these information were written to stdout. To prevent that use

- Windows: \(java\ldots > NUL\)
- Unix: \(java\ldots > /dev/null\)
6.5.16 JMIF Service won't stop
Most likely JMIF runs on a cluster. Connect to the active node and take the corresponding resource offline using cluster administration console. If you bring back JMIF resource online (i.e. starting the service) do not forget to bring depended resources online, too.

6.5.17 JMIF won’t stop
Most likely JMIF runs as service or daemon which is watched by a service wrapper. These tools check the state of running JMIF’s process and restarts it as configured if the process dies. Use the service manager’s stop function to halt JMIF.

6.5.18 JVM Crashed
We noticed that rarely in combination with usage of native external libraries used via JNI. Create a support file (contains the JVM crash dump file) and pass it to us. Try another JVM version and/or vendor.

6.6 T3C
6.6.1 T3C Mask State not updated
In case the mask state is not updated and confirmation on OP is not possible, double check the different address mappings. For performance reasons JMIF fetches all opening’s state at one time. Therefore you may not recognize if JMIF polls an inactive, not existing opening. Check JMIF’s log files for so called injected OP addresses, where JMIF dynamically injects device’s internal addresses into its settings at runtime (memory only, not persisted).
7 HOWTO

7.1 Using Media Converter

Usage of media converter may be recommendable in case of connecting serial devices over Ethernet (long distance, central installation at data center, lack of ports) or to encapsulate devices from broadcast storms.

Basic configuration is usually done by console (serial or telnet in combination with ARP or SNMP) or by proprietary, platform dependent software. We recommend using the serial configuration port (Hyperterminal, TeraTerm or Minicom).

Further configurations are done by terminal (serial/ telnet) or web based (VSCOM only).

The most critical settings are the destination IP address and port, the end of message detection and the serial port parameters.

The amount of serial ports per device should be chosen considering the serial device topology and aspects of redundancy.

7.1.1 Common FAQ

What is the end of message trigger for T3-based machines? There’s no single one! Please choose one of the following options:

- exclusive timeout trigger (e.g. 100ms at 9600 baud)
- add ACK (0x06), ETX (0x03) and SO (0x0E) as end of message tag additionally to the timeout trigger of 100ms (note: most converters treat several delimiters as one expected character sequence and not as different possible separators furthermore T3 expects exactly one datagram per command!)
- use port mode RawTCP Server

What is the end of message trigger for MM2xx/MM5xx based machines? 0xD

What is the end of message trigger for RLP devices? 0xA and a trigger timeout of 5msec.

What are the default serial port settings for MM260 based machines? Baud Rate: 9600, Parity: None, Data Bits: 8, Stop Bit: 1, Flow Control: None. Note: Due to the fact, that the RS pin assignment is proprietary, one has to use an adapter from KARDEX!

Media converter is not ping-able In case all other connections working (local PC to media converter and JMIF server to local PC) most likely the gateway setting on media converter was not done or wrong.

I got no communication - what’s wrong? Go to port statistics of your media converter and check the sent (TX) and received (RX) byte count. Now you should check the settings between PC and converter (TX equals zero) or between device and converter (RX equals zero).

In case of multi port converters (2, 4, 8 or 16) double check configured UDP ports for uniqueness.
After device reset the monitored transfer on serial port is not set to zero - what’s wrong? Most likely there’s a cabling issue on serial port side. Unplug the serial cable, reset device and check the traffic counters again. Plug the cable and check the device’s status after reset anew.

**I got no communication after I changed the port mode - what’s wrong?** Check the COM port settings, sometimes they are linked to the port mode or reset on port mode change.

**I got no communication after I have migrated a JMIF installation - what’s wrong?** Check the destination address of UDP mode’s settings. Most likely your new installation is using a different IP address.

### 7.1.2 VSCOM

The following listing illustrates a typical configuration to access a RLP device over UDP. Communication parameters for serial console are 38400, 8, n, 1, n using null modem cable or adapter. After basic server configuration is done, one could switch from console to Web-Browser using configured IP address and port 80. *There is no need to install drivers or other software!*

```
<table>
<thead>
<tr>
<th>ServerConfig</th>
<th>SerialPorts</th>
<th>Tools</th>
<th>Save&amp;Exit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Parameter</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Server Name</td>
<td>[NETCOM.413]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAC Address</td>
<td>00:04:D9:80:0F:F4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DHCP</td>
<td>Disabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP Address</td>
<td>[192.168.100.30]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Netmask</td>
<td>[255.255.255.0]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Broadcast</td>
<td>[192.168.100.255]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gateway</td>
<td>[192.168.100.18]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DNS</td>
<td>[192.168.100.18]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Domain</td>
<td>[gss-online.com]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ConfigPort</td>
<td>[23]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PrintServerPort</td>
<td>[515]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KeepAlive</td>
<td>Off</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KeepAliveInterval</td>
<td>[0]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```

```
<table>
<thead>
<tr>
<th>Server configuration settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetCom - 413 V2.4.1</td>
</tr>
<tr>
<td>ServerConfig</td>
</tr>
<tr>
<td>-----------------</td>
</tr>
</tbody>
</table>

```

```
Date and Time Settings

Date & Time | [29-04-2010 12:09:59 UTC±0] |

```

Simple Network Time Protocol
7 HOWTO

State | Interval
Mode | IP Address
Interval | [1800 ]
Server | [192.168.100.6 ]

Server configuration settings

NetCom – 413 V2.4.1

ServerConfig | SerialPorts | Tools | Save&Exit |

Serial Settings
Port Nr. | 1
PortType (current) | rs232
DefaultModel | 16550
MaxBandrate | 921600
Model | 16550
Bandrate | 9600
Manual | 9600
FlowType | None
DataBit | 8
Parity | None
StopBit | 1
RxFifoLength | 2048
RxTriggerLevel | [1248 ]
TxFifoLength | 2048
TxTriggerLevel | [800 ]

Serial port configuration settings

Transfer Settings
Port Nr. | 1
Mode | UDP Mode
UDP Port (Local) | [10011 ]
Destination | [de6-00000-000003 ]
UDP Port (Dest) | [10001 ]
UDP MaxPacketSize | [1458 ]
UDP Timeout | [10 ]
UDP Trigger | [\x0D\x0A ]

Serial port configuration settings

Listing 4: NetCom - 413 Setup

69
FAQ

Why are my settings not stored/lost after reboot? Most probably you have set the DIP switches alias operation mode on port 1 to factory settings!

Why connection over crossed ethernet cable for NetCom Manager usage or serial console access may fail? Reset the device to factory defaults using DIP switches and then back port 1 to configuration mode. Do not forget to power off/on the device after changing DIP switches!

What are the DIP switch settings for RLP devices Because RLP devices connected using RS232 and operation mode data port is needed, set \((S1, S2, S3, S4) = (OFF, OFF, ON, ON)\).

7.1.3 MOXA

The following listing illustrates a typical configuration to access a RLP device over UDP. Communication parameters for serial console are 19200, 8, n, 1, n.
### Port Application

<table>
<thead>
<tr>
<th>Port Number</th>
<th>Application</th>
<th>Address Type</th>
<th>Address Range</th>
<th>Port Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TCP Server</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TCP Client</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>UDP Server/Client</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ethernet Modem</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### MOXA NPort Server Express V1.7

```
serverConfig [OP_mode] Serialport Monitor Ping Restart Exit

Config the operation mode of the serial port

ESC: back to menu  Enter: select
```

<table>
<thead>
<tr>
<th>Port Application</th>
<th>More settings</th>
<th>Select for more settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 [UDP Server/Client]</td>
<td>[ ]</td>
<td></td>
</tr>
</tbody>
</table>

### MOXA NPort Server Express V1.7

```
serverConfig [OP_mode] Serialport Monitor Ping Restart Exit

Config the operation mode of the serial port

ESC: back to menu  Enter: select
```

<table>
<thead>
<tr>
<th>Port Application</th>
<th>Begin</th>
<th>End</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 [UDP Server]</td>
<td>[ ]</td>
<td>[ ]</td>
<td>[ ]</td>
</tr>
</tbody>
</table>

### MIF01 Channel Example:

```
Channel32=UDP:192.168.100.32#4001, Port=4001
```

### MOXA NPort Server Express V1.7

```
serverConfig OP_mode [Serialport] Monitor Ping Restart Exit

Config serial port settings

ESC: back to menu  Enter: select
```

<table>
<thead>
<tr>
<th>Port Number</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud Rate (bps)</td>
<td>9600</td>
</tr>
<tr>
<td>Parity</td>
<td>None</td>
</tr>
<tr>
<td>Data Bits</td>
<td>8</td>
</tr>
<tr>
<td>Stop Bit</td>
<td>1</td>
</tr>
<tr>
<td>Flow Control</td>
<td>None</td>
</tr>
<tr>
<td>UART FIFO</td>
<td>Enable</td>
</tr>
</tbody>
</table>

---

71 HOWTO
7 HOWTO

Listing 5: MOXA NPort-211 Setup

7.1.4 Recommended Port Mode

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>T3</td>
<td>UDP/ RawTCP (Server)</td>
</tr>
<tr>
<td>T4</td>
<td>UDP</td>
</tr>
<tr>
<td>MM260</td>
<td>UDP</td>
</tr>
<tr>
<td>MM500</td>
<td>UDP</td>
</tr>
<tr>
<td>RLP</td>
<td>UDP</td>
</tr>
<tr>
<td>Net2</td>
<td>RawTCP (Server)</td>
</tr>
</tbody>
</table>

Table 12: Port Mode

*Note:* In case of mode RawTCP the server’s TCP KeepAlive option should be switched on using the desired keep alive interval. Otherwise an automatic reconnection from JMIF in case of connection lost may not be possible until server reboot.
### 7.1.5 Summary

<table>
<thead>
<tr>
<th>Serial Settings</th>
<th>T3</th>
<th>MM260</th>
<th>RLP</th>
<th>Net2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud Rate</td>
<td>9600</td>
<td>9600</td>
<td>9600</td>
<td>38400</td>
</tr>
<tr>
<td>Data Bits</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Stop Bits</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Parity</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>Even</td>
</tr>
<tr>
<td>Flow Control</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>FIFO</td>
<td>Enable</td>
<td>Enable</td>
<td>Enable</td>
<td>Enable</td>
</tr>
</tbody>
</table>

| Operating or Transfer Settings | | | | |
| Operation Mode | UDP | UDP | UDP | TCP Server or UDP |
| Delimiter/ Trigger | 0x06 | 0x0D | 0x0A | (none) |
| Force Transmit/ Timeout [msec] | 100 | 100 | 10 | 0 |
| Max Connections | 1 | | | |
| TCP Alive Time [sec] | | | 60 or triggered | |
| Destination IP | IP of system hosting JMIF | | | |
| Destination Port | unique port configured at JMIF | | | |
| Local Port | listening port | | | |

Table 13: Generic Serial Device Server Settings Matrix

All other settings should be kept at device server vendor’s factory defaults.

### 7.2 Connecting RLP Devices

RLP devices are loosely coupled with the machine (power only). They are controlled using one serial port per device. Therefore the usage of media converters is strongly recommended.

One could use virtual COM ports based on media converter’s drivers (serial port tunneling over Ethernet). But instead of this we recommend the usage of raw UDP. Raw TCP should work as well but is rarely used. Do not forget to

- adjust the port mode (common factory default: driver mode)!
- activate the listener mode on UDP channels!

### 7.2.1 Basic Tests

Before starting any kind of operations do the following tests:
• switch off the machine and check whether the RLP swing into the plumb-line
• switch on the machine, check whether the RLP drives the initial positions and afterwards is perpendicularly stopped

If one of the tests fails contact your KARDEX technician.

7.2.2 Configuration and Calibrating

Configuration is done globally or per device while missing local values are replaced by global ones. The data for carrier length and width as well as the installation height (fixing) per opening are needed.

To avoid manual calibration JMIF supports since version 1.5.30 automatic y-offset corrections on standard tray types 610, 813 and 864. Omit settings \( x\text{Offset} \) and \( y\text{Offset} \) and just use \( c\text{Length} \) and \( c\text{Width} \).

In case of non standard tray width or linear deviation a calibration is required. For that purpose the reference position \((x, y, z) = (1098, 432, 0)\) has to be used as

• default for offsets and independent from carrier width
• for a test request itself

Measure the real distance between origin (left front corner) and the illuminated position. Use the measured values as new offsets.

If there are non-linear deviations contact your KARDEX technician. If the light spot’s diameter is too big, consider using the laser head.

The \( z \) and \( z1 \) offsets could be used in case of machine type does not support reading level of tray’s level inside the access opening.

Permanent glow is supported by JMIF but there is no guaranty that the RLP not caused damage to except of low level encoder step mode is used. In case action \texttt{light} does not work in default mode, enable the initialization, i.e. set \texttt{skipInit} to false. To omit axis initialization only use \texttt{skipInitAxis}.

If there are no millimeter coordinates available JMIF supports several methods to calculate them from the given request’s data: linear scaling, mapping and even script or expression based calculation at runtime.

7.2.3 RLP With Embedded LAN Interface

Since mid of 2015 RLP devices will be delivered with an embedded LAN interface (Lantronix XPort Direct+). There are several known issues configuring the module using \texttt{DeviceInstaller} tool or \texttt{Web Manager}. In case of using the \texttt{DeviceInstaller} ensure that you are directly connected to the LAN interface or at least to machine’s switch but without having an up-link activated. In case of \texttt{Web Manager} ensure that most current device Firmwares (ROM and most important COB) is installed because of Web Browser incompatibility issues.

\texttt{Warning: In case of usage of incompatible tools or invalid configuration settings the network interface port may become unavailable! Because there’s no hardware factory reset switch nor a second port you may have to exchange the complete module.}

If the module’s IP is set the channel setup could be easily done with Telnet to port 9999. Channel settings (some are bit-coded hex values):

• Baudrate: 9600
7 HOWTO

- I/F Mode: 4C (RS-232C, 8-bit, No Parity, 1 stop bit)
- Flow: 00 (No flow control)
- Port No: 4011 (local port)
- ConnectMode: CC (Directed UDP)
- Datagram Type: 01 (directed or broadcast)
- Send as Broadcast: N
- Remote IP Address: (nnn).(nnn).(nnn).(nnn)
- Remote Port: 4011
- Pack Ctrl: 00 (Packing Interval: 12msec, Trailing Characters: None, 2-Byte Send Character Sequence: No, Send Immediately After Send chars: No)
- SendChar 1: 0
- SendChar 2: 00 (disabled)

7.3 Connecting TIC Device

7.3.1 Naming

TIC stands for Transaction Information Center, sometimes also called TIC (light) matrix (display), dot matrix board or (strip) position indicator. JMIF connects to TIC using TIC’s component EIM (Ethernet Interface Module).

7.3.2 Configuration

The TIC device could be used stand alone or as sub system. Main configuration is done by setting up the display formatting:

- Module: hardware part of TIC device (a TIC module), amount of modules could be configured but is detected by JMIF as default
- Segment: part of the whole text to display, position could be changed due to lack of space, maps to display formatter’s lines, one up to three segments are supported
- Control Characters: used to show arrows or toggle text blinking, could be used anywhere inside display formatter’s line spec or directly from host (stand alone mode recommended)
  - %ˆ% arrow up
  - %|% arrow down
  - %B% toggle blinking
  - %B % arrow up blinking
  - %B|% arrow down blinking
- Positioning: either based on pos and depth using scaling or more accurately (max. ±5mm tolerance) on millimeter values given by the info fields x and y
- Offset: (left) column offset in millimeter
- Bit Maps: used to replace all non US-ASCII characters, bit map font designer for Windows available on demand
7.3.3 FAQ

**I want to use TIC directly without given carrier**  Carrier zero is used to clear the display. All other values have no meaning. But if the carrier field is undefined it’s default value is zero! Therefore any given numeric field could be reused as carrier field, e.g. the code field.

**Scaling factor**  In case only raster position coordinates are given based on position lamps or Vario indices, one could use a value of 50, which is the lowest distance between two position lamp or Vario segments.

**TIC does not respond on 4001 but Web UI is accessible**  Check whether the modules are connected properly and listed inside Web UI.

**TIC seems to be crashed**  TIC allows only one client connection. If the client connection was not closed properly a new one will not be accepted. JMIF detects this situations and resolves it automatically.

**TIC crashed**  TIC may crash if illegal characters or bit maps were sent. Check message filtering, the display formatters ASCII mode setup and bit maps used.

**Communication timeouts**  Response time depends on TIC length. A TCP timeout value of 2000 msec should be sufficient.
7.4 C3000 Position Indicator Vario

C3000 Position Indicator Vario (called Vario) is used to display pick/store information near to the tray itself. It is available as dot light, 2-digit 7-segment numerical display or 1-digit 7-segment alphanumeric display.

Commonly 2-digit 7-segment numerical display is used where display in the range of pos and pos2 show the value given in depth, e.g. 00 − 00 − 02 − 02 − 00 − 00 − 00 using position (3, 4) and depth 2 or (2, 2).

There are some common reasons, that Vario do not work properly:

1. Machine Parameter Setup: Vario setup was not done, if all segments permanently glow (showing 00 or 0 in case of 7-segment display). Contact your local KARDEX service department.
2. Invalid Addressing: the position and depth addresses are 1-based. In case of pos2 and depth2 are used, values must be equal or greater than the from value. In case of only one value is out of range, nothing will be displayed.
3. Firmware: new extended display command is not working properly. Use JMIF’s forcedVersion parameter with hexadecimal value 10900.

7.5 Dual Tray

7.5.1 Basics

The dual tray option on machine side (also called multi-tray presentation) has to be licensed and activated. Check device’s Operating Manual! It is only recommend in case of

- one opening or time based exclusive opening usage (shift operation)
- uniform location high on all trays
- carrier next data (tray, which should be prepared on the lift) already known at the time, when the previous carrier is requested or order based workflow
- in combination with a RLP device the upper level plus the location high is within the RLP device’s specification.

JMIF tries to switch a position on the fetched tray on demand as well as allow confirmation while tray preparation is in progress. But there’s for technical reasons no guaranty that the tray will be prepared on the lift. The response code belongs in any case to the carrier requested to the opening! The level (upper/lower) is chosen by JMIF.

In case host system wants to toggle dual tray usage on demand, host system could switch off dual tray usage using same values on fields carrier and carrierNext.

For usage on put operations which potentially increase location’s/ tray’s height it’s recommend to indicate movement type with request and use option “denyDualTrayOnStore”.

Look ahead for carrier to be prepare on lift could also be done by JMIF. For this purpose request queuing per access opening has to be used. Look ahead depth is configurable.

7.5.2 How is it working?

Dual tray enhance tray delivery time by preparing next tray on machine’s lift while operator is working on the carrier moved into the opening. Figures 18 and 19 illustrate a sample workflow (cross section view).
Figure 17: Dual Tray

Figure 18: Dual Tray Sample
<table>
<thead>
<tr>
<th>No</th>
<th>Address</th>
<th>Carrier</th>
<th>Carrier Next</th>
<th>Position</th>
<th>Depth</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>06</td>
<td>01 003 004</td>
<td>3 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07</td>
<td>01 003 004</td>
<td>3 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>08</td>
<td>01 004 005</td>
<td>2 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>09</td>
<td>01 004 005</td>
<td>2 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>01 005 000</td>
<td>1 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>01 005 000</td>
<td>2 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>01 005 000</td>
<td>2 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>01 000 000</td>
<td>0 0</td>
<td>Opening will be cleared.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 19: Dual Tray Sample (cont.)
7.6 Modifying Support File Creation Behaviour

A support file contains all files from JMIF application folder needed to analyse an incident. Default file name filter is set to a local subfolder called transfer as well as the file name extensions log, properties, conf, xml, xslt and js. Additional external folders could be added using the setting jmif.supportdirs.

In case ftp access is possible one could configure jmif.support.ftpHostname, jmif.support.ftpUsername and jmif.support.ftpPassword to automatically upload created support files. JMIF tries to created a subfolder using the order number from license information to store the files. Because logging is temporarily disabled during such a support request, there will be no feedback about success or failures.

7.7 Firewalling

If communication between host system and JMIF or JMIF and hardware devices is between different departments or even over a WAN it may be necessary to configured existing Firewalls between involved systems. That also applies to existing and activated local firewalls. Table 14 lists some common rules.

<table>
<thead>
<tr>
<th>Communication Partner</th>
<th>Direction</th>
<th>Protocol</th>
<th>Port</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>JMIF internal</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JMIF Application Lock</td>
<td>IN</td>
<td>TCP</td>
<td>52525</td>
<td>configurable</td>
</tr>
<tr>
<td>Service Wrapper/ Win32</td>
<td>IN</td>
<td>TCP</td>
<td>32000+</td>
<td>configurable, loopback, JVM comm.</td>
</tr>
<tr>
<td>Host System/ JMIF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JMIF as TCP server</td>
<td>IN</td>
<td>TCP</td>
<td>xxxxx</td>
<td>configured port</td>
</tr>
<tr>
<td>JMIF as TCP client</td>
<td>OUT</td>
<td>TCP</td>
<td>xxxxx</td>
<td>configured port</td>
</tr>
<tr>
<td>JMIF/ Devices</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C3000 PLC</td>
<td>OUT</td>
<td>TCP</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td>C3000 OP</td>
<td>OUT</td>
<td>TCP</td>
<td>80</td>
<td>OPbrowser tool</td>
</tr>
<tr>
<td>C2000 PLC</td>
<td>IN, OUT</td>
<td>UDP</td>
<td>2002</td>
<td>JMIF needs unique port per PLC</td>
</tr>
<tr>
<td>C2000 PLC</td>
<td>OUT</td>
<td>TCP</td>
<td>23</td>
<td>service tool</td>
</tr>
<tr>
<td>MM260 AO</td>
<td>IN, OUT</td>
<td>UDP</td>
<td>2003</td>
<td>JMIF needs unique port per AO</td>
</tr>
<tr>
<td>Intertex Web UI</td>
<td>OUT</td>
<td>TCP</td>
<td>80</td>
<td>see table 13</td>
</tr>
<tr>
<td>Intertex PLC</td>
<td>OUT</td>
<td>TCP</td>
<td>4xx0</td>
<td>xx = unit no.</td>
</tr>
<tr>
<td>RLP Web UI</td>
<td>OUT</td>
<td>TCP</td>
<td>80</td>
<td>see table 13</td>
</tr>
<tr>
<td>RLP PLC</td>
<td>IN, OUT</td>
<td>UDP</td>
<td>4xx</td>
<td>xx = unit no., y = AO no.</td>
</tr>
<tr>
<td>RLP Console</td>
<td>OUT</td>
<td>TCP</td>
<td>23, 9999</td>
<td>23 common, 9999 Lantronix</td>
</tr>
<tr>
<td>TIC EIM Web UI</td>
<td>OUT</td>
<td>TCP</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>TIC EIM PLC</td>
<td>OUT</td>
<td>TCP</td>
<td>4001</td>
<td></td>
</tr>
</tbody>
</table>

Table 14: Firewall Rules
7.8 Building a JMIF Appliance

A JMIF appliance based on Debian Linux (version 8 jessie) optionally as virtual machine could be quickly build following these steps:

1. Create a virtual machine based on Linux Debian 8, 64bit, with 1024MB RAM and a 10GB vdisk or use appropriate hardware (no desktop/core: 512MB RAM and 2GB vdisk).
2. Boot from Debian’s netinst image.
3. Select software collections “SSH server” and “standard system tools” (core) and a desktop environment (desktop variant). In case you prefer the Lightweight X11 Desktop Environment only do not select a desktop environment here but just use “apt-get install lxde-core gdm” later and install some useful tools using “apt-get install xpdf leafpad iceweasel filezilla”.
4. On VMware’s hypervisor products run “apt-get install open-vm-tools” and “apt-get install open-vm-tools-desktop” for drag’s drop support on desktop. You need to add the contrib archive to your apt source settings.
5. Create a standard user account “jmif” with a home directory to be used in next steps.
6. Install JMIF by uploading JMIF’s tgz file and extract it using “tar xvzf”.
7. Install default JDK using “apt-get install default-jdk” or a private JRE using a (t)gz file from Oracle.
8. Configure and start JMIF as needed. In case you use the init system do not forget to run “update-rc.d jmif defaults”.
9. Optionally download and extract TeamViewer QS for Linux.
References


Legal Notice

© Kardex Software GmbH, Im Bruch 2, 76744 Wörth a.R., Germany.

All rights reserved worldwide.

This document must not be reproduced, revised, rewritten or converted into a computer language or any other language in any form or for any purpose using electronic, mechanical, magnetic, optical or any other means, whether in part or in full, unless express written permission has first been obtained.

Unless authorised by Kardex Software GmbH, any attempt to reproduce or distribute this manual or to store its contents on data carriers shall be classed as an infringement of copyright and will result in prosecution.

Product names are quoted for informational purposes only, which by no means constitutes a misuse of trademarks.
<table>
<thead>
<tr>
<th>Ed.</th>
<th>Date</th>
<th>Au.</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4.0</td>
<td>2008-10-20/29</td>
<td>db</td>
<td>Amendment of JMIF’s manual from 2005</td>
</tr>
<tr>
<td>1.4.1</td>
<td>2008-10-31</td>
<td>db</td>
<td>Float enhancements</td>
</tr>
<tr>
<td>1.4.2</td>
<td>2009-01-27</td>
<td>db</td>
<td>Corrections</td>
</tr>
<tr>
<td>1.4.3</td>
<td>2009-03-04</td>
<td>db</td>
<td>Compiler warnings eliminated, more corrections, transformer infos added</td>
</tr>
<tr>
<td>1.4.4</td>
<td>2009-03-11</td>
<td>db</td>
<td>Naming conventions, regular expressions section and dispatcher table added</td>
</tr>
<tr>
<td>1.4.5</td>
<td>2009-04-14/17</td>
<td>db</td>
<td>Minor updates, FAQs added, known standard fields added</td>
</tr>
<tr>
<td>1.4.6</td>
<td>2009-11-17</td>
<td>db</td>
<td>Chainsaw port setting note added, mixed mode workflow informations added</td>
</tr>
<tr>
<td>1.5.0</td>
<td>2010-01-12</td>
<td>db</td>
<td>Updates regarding JVM, FAQs OutOfMemory and Proxy infos added</td>
</tr>
<tr>
<td>1.5.1</td>
<td>2010-03-03</td>
<td>db</td>
<td>Update of supported machine types, information about regular expressions added, FAQ Windows Versions added</td>
</tr>
<tr>
<td>1.5.2</td>
<td>2010-04-29</td>
<td>db</td>
<td>Sections RLP and HowTo/Mediaconverter added, FAQ Hide Console Window added</td>
</tr>
<tr>
<td>1.5.3</td>
<td>2010-05-11</td>
<td>db</td>
<td>Advanced information for JMX added, FAQ Crypted values added, minor tweaks</td>
</tr>
<tr>
<td>1.5.4</td>
<td>2010-05-12</td>
<td>db</td>
<td>FAQ JDBC connections added</td>
</tr>
<tr>
<td>1.5.5</td>
<td>2010-07-12</td>
<td>db</td>
<td>FAQ C3000 Timeout Troubleshooting added</td>
</tr>
<tr>
<td>1.5.6</td>
<td>2010-07-14</td>
<td>db</td>
<td>XMPP channel added</td>
</tr>
<tr>
<td>1.5.7</td>
<td>2010-07-21</td>
<td>db</td>
<td>FAQs and HOWTOs extensions</td>
</tr>
<tr>
<td>1.5.8</td>
<td>2010-10-18</td>
<td>db</td>
<td>Intertex, streamed channels, special packages, demo mode, major upgrade</td>
</tr>
<tr>
<td>1.5.9</td>
<td>2010-10-21</td>
<td>db</td>
<td>infos about Lilith, RawTCP and extended location data added</td>
</tr>
<tr>
<td>1.5.10</td>
<td>2010-10-26</td>
<td>db</td>
<td>FAQ added</td>
</tr>
<tr>
<td>1.5.11</td>
<td>2010-10-27</td>
<td>db</td>
<td>some more FAQs added</td>
</tr>
<tr>
<td>1.5.12</td>
<td>2010-11-12</td>
<td>db</td>
<td>standard’s order types description added</td>
</tr>
<tr>
<td>1.5.13</td>
<td>2010-11-12</td>
<td>db</td>
<td>notes on NT service support added</td>
</tr>
<tr>
<td>1.5.14</td>
<td>2010-11-25</td>
<td>db</td>
<td>FAQs reorganized, FAQ for RLP and rational numbers added</td>
</tr>
<tr>
<td>1.5.15</td>
<td>2010-12-07</td>
<td>db</td>
<td>Note about usage of MSQL Integrated Authentication added</td>
</tr>
<tr>
<td>1.5.16</td>
<td>2010-12-08</td>
<td>db</td>
<td>Media converter FAQ extended</td>
</tr>
<tr>
<td>1.5.17</td>
<td>2010-12-09</td>
<td>db</td>
<td>Minor addition</td>
</tr>
<tr>
<td>1.5.18</td>
<td>2010-12-13</td>
<td>db</td>
<td>media converter information extended, minor updates on standard and FAQ section, FAQ return code mapping added</td>
</tr>
<tr>
<td>1.5.19</td>
<td>2010-12-15</td>
<td>db</td>
<td>TIC FAQ added</td>
</tr>
<tr>
<td>1.5.20</td>
<td>2010-12-21</td>
<td>db</td>
<td>TIC configuration basics added, default values for RegExTelegram items</td>
</tr>
<tr>
<td>1.5.21</td>
<td>2010-12-22</td>
<td>db</td>
<td>new FAQ subsection security added</td>
</tr>
<tr>
<td>1.5.22</td>
<td>2010-12-22</td>
<td>db</td>
<td>Work-flow mode 1 notes added, minor updates</td>
</tr>
<tr>
<td>1.5.23</td>
<td>2011-01-11</td>
<td>db</td>
<td>FAQ about NAK added, FAQ JDBC updated</td>
</tr>
<tr>
<td>1.5.24</td>
<td>2011-01-11</td>
<td>db</td>
<td>FAQ Reset Log File added</td>
</tr>
</tbody>
</table>

Continued on Next Page...
<table>
<thead>
<tr>
<th>Ed.</th>
<th>Date</th>
<th>Au.</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5.25</td>
<td>2011-01-20</td>
<td>db</td>
<td>FAQ FTP Server added</td>
</tr>
<tr>
<td>1.5.26</td>
<td>2011-02-17</td>
<td>db</td>
<td>FAQ supported encodings added, org logo update</td>
</tr>
<tr>
<td>1.5.27</td>
<td>2011-03-01</td>
<td>db</td>
<td>CI update, media converter HowTo updates</td>
</tr>
<tr>
<td>1.5.28</td>
<td>2011-04-07</td>
<td>db</td>
<td>DUMP command, FAQ File vs. TCP channel added</td>
</tr>
<tr>
<td>1.5.29</td>
<td>2011-04-13</td>
<td>db</td>
<td>FAQs regarding undefined destination and work file creation error added</td>
</tr>
<tr>
<td>1.5.30</td>
<td>2011-04-14</td>
<td>db</td>
<td>FAQ dual tray added</td>
</tr>
<tr>
<td>1.5.31</td>
<td>2011-04-18</td>
<td>db</td>
<td>typesetting enhanced, cyclic typesetting system update, uninstall sub section added, FAQ about telegram format added</td>
</tr>
<tr>
<td>1.5.32</td>
<td>2011-04-26</td>
<td>db</td>
<td>FAQ Starting JMIF as a Service fails added, JMIF paused caused by corrupted log file added</td>
</tr>
<tr>
<td>1.5.33</td>
<td>2011-06-27</td>
<td>db</td>
<td>order type queue state added</td>
</tr>
<tr>
<td>1.5.34</td>
<td>2011-07-12</td>
<td>db</td>
<td>note on possible return codes of alive telegram added</td>
</tr>
<tr>
<td>1.5.35</td>
<td>2011-09-06</td>
<td>db</td>
<td>notes about supported channels and devices inside standard added, Dual Tray HowTo added</td>
</tr>
<tr>
<td>1.5.36</td>
<td>2011-10-25</td>
<td>db</td>
<td>sections on includes and property key sequence added</td>
</tr>
<tr>
<td>1.5.37</td>
<td>2011-10-26</td>
<td>db</td>
<td>typos, MM2xx FAQs added</td>
</tr>
<tr>
<td>1.5.38</td>
<td>2011-11-04</td>
<td>db</td>
<td>alive telegram return code sample added</td>
</tr>
<tr>
<td>1.5.39</td>
<td>2011-11-11</td>
<td>db</td>
<td>FAQs on NAK and DirectStore timeouts added; typos fixed</td>
</tr>
<tr>
<td>1.5.40</td>
<td>2011-11-14</td>
<td>db</td>
<td>Dispatcher and package lists updated</td>
</tr>
<tr>
<td>1.5.41</td>
<td>2011-11-28</td>
<td>db</td>
<td>JMIF Service won’t stop FAQ added</td>
</tr>
<tr>
<td>1.5.42</td>
<td>2011-11-30</td>
<td>db</td>
<td>error code table completed</td>
</tr>
<tr>
<td>1.5.43</td>
<td>2012-01-10</td>
<td>db</td>
<td>added running JMIF as Linux daemon, building a JMIF appliance, FAQ JMIF PID, FAQ JMIF won’t stop; extended FAQ Starting fails</td>
</tr>
<tr>
<td>1.5.44</td>
<td>2012-01-13</td>
<td>db</td>
<td>Hide JMIF’s Loader Output FAQ added</td>
</tr>
<tr>
<td>1.5.45</td>
<td>2012-01-16</td>
<td>db</td>
<td>MM2xx Manual Memory Usage and Display FAQ added</td>
</tr>
<tr>
<td>1.5.46</td>
<td>2012-01-27</td>
<td>db</td>
<td>minor notes about RLP usage added</td>
</tr>
<tr>
<td>1.5.47</td>
<td>2012-02-28</td>
<td>db</td>
<td>minor notes about running in x64 JVM added</td>
</tr>
<tr>
<td>1.5.48</td>
<td>2012-03-06</td>
<td>db</td>
<td>notes about migrating JMIF installation added (licensing and media converter usage)</td>
</tr>
<tr>
<td>1.5.49</td>
<td>2012-03-20</td>
<td>db</td>
<td>FAQ Log Files Missing or Outdated added</td>
</tr>
<tr>
<td>1.5.50</td>
<td>2012-04-12</td>
<td>db</td>
<td>Notes about SQL scripts and tray table sync cron job added</td>
</tr>
<tr>
<td>1.5.51</td>
<td>2012-06-01</td>
<td>db</td>
<td>Section about incompatibly changed settings added, T3 delimiter stuff clarified</td>
</tr>
<tr>
<td>1.5.52</td>
<td>2012-06-08</td>
<td>db</td>
<td>C3000 Position Indicator Vario HowTo added</td>
</tr>
<tr>
<td>1.5.53</td>
<td>2012-06-19</td>
<td>db</td>
<td>C3000 character encoding notes added</td>
</tr>
<tr>
<td>1.5.54</td>
<td>2012-09-28</td>
<td>db</td>
<td>MM2xx, RLP and media converter FAQs updated, minor tweaks</td>
</tr>
<tr>
<td>1.5.55</td>
<td>2012-09-28</td>
<td>db</td>
<td>correction: “local JRE” replaced by “private JRE”</td>
</tr>
<tr>
<td>1.5.56</td>
<td>2012-10-22</td>
<td>db</td>
<td>C2000/MM2xx timeout FAQ extended, C2000 position lamps FAQ added</td>
</tr>
</tbody>
</table>

Continued on Next Page...
<table>
<thead>
<tr>
<th>Ed.</th>
<th>Date</th>
<th>Au.</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5.58</td>
<td>2012-11-13</td>
<td>db</td>
<td>C3000 and T3C FAQs added</td>
</tr>
<tr>
<td>1.5.59</td>
<td>2012-11-20</td>
<td>db</td>
<td>FAQ user device input handling added</td>
</tr>
<tr>
<td>1.5.60</td>
<td>2012-12-19</td>
<td>db</td>
<td>Error and Event Codes table revised</td>
</tr>
<tr>
<td>1.5.61</td>
<td>2013-01-22</td>
<td>db</td>
<td>legal folder description added, code 13 added, minor tweaks</td>
</tr>
<tr>
<td>1.5.62</td>
<td>2013-06-06</td>
<td>db</td>
<td>DIWS channel added</td>
</tr>
<tr>
<td>1.6.0</td>
<td>2013-06-06</td>
<td>db</td>
<td>JRE minimum requirements changed to version 1.6</td>
</tr>
<tr>
<td>1.6.1</td>
<td>2013-07-05</td>
<td>db</td>
<td>device list revised, RLP configuration notes added</td>
</tr>
<tr>
<td>1.6.2</td>
<td>2013-07-22</td>
<td>db</td>
<td>a media converter FAQ added</td>
</tr>
<tr>
<td>1.6.3</td>
<td>2013-07-23</td>
<td>db</td>
<td>notes about reverse mapping and field data types</td>
</tr>
<tr>
<td>1.6.4</td>
<td>2013-08-01</td>
<td>db</td>
<td>JVM crash dump FAQ added</td>
</tr>
<tr>
<td>1.6.5</td>
<td>2013-08-28</td>
<td>db</td>
<td>T3 timing and Intertex FAQs added, dual tray basics reworked, typos fixed, obligatory Legal Notice added</td>
</tr>
<tr>
<td>1.6.6</td>
<td>2013-10-07</td>
<td>db</td>
<td>Modifying Support File Creation Behaviour HowTo added</td>
</tr>
<tr>
<td>1.6.7</td>
<td>2013-11-11</td>
<td>db</td>
<td>MM260 serial connection FAQ added</td>
</tr>
<tr>
<td>1.6.8</td>
<td>2013-12-17</td>
<td>db</td>
<td>some more well known C2000 timeout reasons added</td>
</tr>
<tr>
<td>1.6.9</td>
<td>2014-01-10</td>
<td>db</td>
<td>Interfaces and Dispatchers table updated</td>
</tr>
<tr>
<td>1.6.10</td>
<td>2014-02-28</td>
<td>db</td>
<td>FAQs Switch Port Security and WinNT 6.2 and 6.3 Tips and Tricks added</td>
</tr>
<tr>
<td>1.6.11</td>
<td>2014-04-08</td>
<td>db</td>
<td>notes about Java properties file format added</td>
</tr>
<tr>
<td>1.6.12</td>
<td>2014-04-11</td>
<td>db</td>
<td>compatibility note added, table design re-worked</td>
</tr>
<tr>
<td>1.6.13</td>
<td>2014-04-24</td>
<td>db</td>
<td>Generic Serial Device Server Settings Matrix added</td>
</tr>
<tr>
<td>1.6.14</td>
<td>2014-07-03</td>
<td>db</td>
<td>Query Extended Storage Location Information FAQ clarified</td>
</tr>
<tr>
<td>1.6.15</td>
<td>2014-07-09</td>
<td>db</td>
<td>RLP deviation on Y-axis FAQ added</td>
</tr>
<tr>
<td>1.6.16</td>
<td>2014-07-17</td>
<td>db</td>
<td>dispatcher table updated</td>
</tr>
<tr>
<td>1.6.17</td>
<td>2014-07-25</td>
<td>db</td>
<td>C3000 acts strange and showing mysterious texts FAQ added</td>
</tr>
<tr>
<td>1.6.18</td>
<td>2014-09-22</td>
<td>db</td>
<td>order type code 63 added</td>
</tr>
<tr>
<td>1.6.19</td>
<td>2014-10-15</td>
<td>db</td>
<td>Firewalling HowTo added</td>
</tr>
<tr>
<td>1.6.20</td>
<td>2014-10-30</td>
<td>db</td>
<td>note about support scripts added; chapter Quick Migration Steps added; Media Converter FAQ added</td>
</tr>
<tr>
<td>1.6.21</td>
<td>2014-11-04</td>
<td>db</td>
<td>FAQ C3000 Measurement Units added</td>
</tr>
<tr>
<td>1.6.22</td>
<td>2015-01-19</td>
<td>db</td>
<td>FAQ RLP device sporadically encounters an obstacle</td>
</tr>
<tr>
<td>1.6.23</td>
<td>2015-01-20</td>
<td>db</td>
<td>Response messages undeliverable on TCP connections refined, Incompatibly Changed Settings entry added</td>
</tr>
<tr>
<td>1.6.24</td>
<td>2015-02-02</td>
<td>db</td>
<td>Incompatibly Changed Settings section updated</td>
</tr>
<tr>
<td>1.6.25</td>
<td>2015-03-20</td>
<td>db</td>
<td>DirectStore timeouts figure added</td>
</tr>
<tr>
<td>1.6.26</td>
<td>2015-03-26</td>
<td>db</td>
<td>FAQs for NT services and media converter extended</td>
</tr>
<tr>
<td>1.6.27</td>
<td>2015-07-07</td>
<td>db</td>
<td>TIC FAQs added; typos fixed</td>
</tr>
<tr>
<td>1.6.28</td>
<td>2015-10-27</td>
<td>db</td>
<td>note about JRE/Windows XP support added</td>
</tr>
<tr>
<td>1.6.29</td>
<td>2015-11-06</td>
<td>db</td>
<td>JMIF appliance building updated, Firewall rules table enhanced</td>
</tr>
<tr>
<td>1.6.30</td>
<td>2015-12-07</td>
<td>db</td>
<td>section RLP With Embedded LAN Interface added, TIC communication timeout FAQ added, firewall rules table updated</td>
</tr>
</tbody>
</table>

Continued on Next Page…
<table>
<thead>
<tr>
<th>Ed.</th>
<th>Date</th>
<th>Au.</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.6.31</td>
<td>2016-02-09</td>
<td>db</td>
<td>link coloring re-worked, further order type codes added</td>
</tr>
<tr>
<td>1.6.32</td>
<td>2016-03-02</td>
<td>db</td>
<td>FAQ T3 Timeout Issues added</td>
</tr>
<tr>
<td>1.6.33</td>
<td>2016-07-15</td>
<td>db</td>
<td>Serial Device Server Settings Matrix/Net2 fixed</td>
</tr>
<tr>
<td>1.6.34</td>
<td>2016-07-22</td>
<td>db</td>
<td>RegEx Online Reference list extended</td>
</tr>
<tr>
<td>1.6.25</td>
<td>2016-10-11</td>
<td>db</td>
<td>DirectStore’s timeout description fixed</td>
</tr>
<tr>
<td>1.6.25</td>
<td>2016-11-11</td>
<td>db</td>
<td>user input FAQ re-worked, properties listings beautify</td>
</tr>
<tr>
<td>1.6.26</td>
<td>2016-12-08</td>
<td>db</td>
<td>WFM 5 added</td>
</tr>
<tr>
<td>1.6.27</td>
<td>2016-12-14</td>
<td>db</td>
<td>device list and WFM 1 corrections</td>
</tr>
<tr>
<td>1.6.28</td>
<td>2017-05-11</td>
<td>db</td>
<td>FAQ about JRE security and updates added</td>
</tr>
<tr>
<td>1.6.29</td>
<td>2017-07-19</td>
<td>db</td>
<td>Firewalling table updated</td>
</tr>
<tr>
<td>1.6.30</td>
<td>2017-07-20</td>
<td>db</td>
<td>nohup FAQ added</td>
</tr>
<tr>
<td>1.6.31</td>
<td>2017-07-21</td>
<td>db</td>
<td>DISPLAY FAQ added, list styling issues fixed</td>
</tr>
<tr>
<td>1.6.32</td>
<td>2017-08-01</td>
<td>db</td>
<td>JRE security and update FAQ extended</td>
</tr>
<tr>
<td>1.6.33</td>
<td>2017-08-02</td>
<td>db</td>
<td>Code table updated</td>
</tr>
<tr>
<td>1.6.34</td>
<td>2017-12-04</td>
<td>db</td>
<td>channel list updated</td>
</tr>
<tr>
<td>1.6.35</td>
<td>2017-12-14</td>
<td>db</td>
<td>JBDC/ Oracle FAQ extended</td>
</tr>
<tr>
<td>1.6.36</td>
<td>2018-02-09</td>
<td>db</td>
<td>some dual tray notes added</td>
</tr>
<tr>
<td>1.6.37</td>
<td>2018-02-13</td>
<td>db</td>
<td>FAQ Prevent useless carrier move back operations added, section Getting Started updated</td>
</tr>
<tr>
<td>1.6.38</td>
<td>2018-03-05</td>
<td>db</td>
<td>Log File Analysis FAQ updated</td>
</tr>
<tr>
<td>1.6.39</td>
<td>2018-04-30</td>
<td>db</td>
<td>quotation mark usage unified</td>
</tr>
<tr>
<td>1.6.40</td>
<td>2018-05-07</td>
<td>db</td>
<td>RLP: Configuration and Calibrating reworked</td>
</tr>
<tr>
<td>1.6.41</td>
<td>2018-23-07</td>
<td>db</td>
<td>FAQ turned position lamps extended</td>
</tr>
</tbody>
</table>
We're always happy to receive constructive criticism.

typeset by l\TeX