

# **JI Integration**

## **Supplemental Reference Guide**

Version 4.5

January 2025  
(originally released December 2004)

---

This document applies to JI Integration Version 4.5 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 1999–2025 Software GmbH, Darmstadt, Germany and/or their licensors.

The name Software GmbH and all Software GmbH product names are either trademarks or registered trademarks of Software GmbH and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Use of this software is subject to adherence to Software GmbH's licensing conditions and terms. These terms are part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices and license terms, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products." This document is part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

**Document ID: JI-SUPP-RF-45-20250131**

---

# Table of Contents

## About this Guide 5

Before You Begin .....	5
Organization .....	5
Formatting Conventions .....	6
Documentation Set .....	6
Viewing the Documentation Online .....	8

## Chapter 1. JI Integration Commands .....9

ea_admin .....	10
ea_cfgmgr .....	14
ea_convertldb .....	16
ea_convertmap .....	19
ea_createdb .....	20
ea_dbping .....	22
ea_dbshutdown .....	24
ea_dbsvr .....	26
ea_envmgr .....	29
ea_exportdb .....	32
ea_importdb .....	35
ea_mapmaker .....	38
ea_mapplayer .....	55
ea_ping .....	57
ea_ressvr .....	60
ea_shutdown .....	62
ea_shutsrvr .....	66
ea_start .....	68
ea_status .....	72
ea_sysmon .....	75
ji_svc .....	77

## Chapter 2. JI Integration GBBasic Reference .....81

## Chapter 3. JI Integration EAServiceBean Reference .....83

## Chapter 4. JClient3 API Reference .....85

## Chapter 5. JI Integration Error Messages .....87

## Chapter 6. Keyboard Mapping .....103

Setting the Keyboard Map File in MapMaker. ....	103
TN3270 Keyboard Mapping .....	104

TN5250 Keyboard Mapping .....	107
Key Map File Structure .....	110
Emulation Function Names .....	110
Keyboard Sequences .....	113
Examples .....	114
Control Keys .....	114
Other Keys .....	114
Protocol Agent Terminal Keyboard Mapping .....	114
Keyboard Definition .....	115
Key Map File Structure .....	118
TN3270 Emulation .....	119
TN5250 Emulation .....	121
Telnet Emulation .....	123
Keyboard Sequences .....	124
Control Keys .....	125
Escape Sequence .....	125
Other Keys .....	126
Application Keypad in Telnet Mode .....	127
101 Key Keyboard .....	127
84 Key Keyboard .....	128
The Terminfo File .....	128
 <b>Chapter 7. Configuration Files .....</b>	<b>131</b>
Order of Precedence for Command Line/Configuration File Parameters .....	132
Windows Directory Names .....	133
The Environment Manager Configuration File .....	134
The Resource Server Configuration File .....	139
The Configuration Manager Configuration File .....	145
The System Monitor Configuration File .....	147
Environment Settings .....	148
Environment Variables .....	148
The environment.ccf Configuration File .....	149
 <b>Chapter 8. Logging Information .....</b>	<b>151</b>
Log File Location .....	151
Using the System Monitor for Event Logging .....	151
Environment Manager Logging .....	151
Setting the Log Level .....	152
Environment Manager Log Levels .....	152
Setting the Name and/or Location of the Log File .....	153
Information Logged to the Environment Manager Log File .....	154
JCluster Logging .....	154
JCluster Log File Name .....	155
Information Logged to the JCluster Log File .....	156
JService Logging .....	156
Setting the JService Log Level .....	157
JService Log Levels .....	157

JService Log File Name .....	158
Dynamically Setting JCluster and JService Logging .....	158
Dynamically Changing the JCluster Trace Level .....	159
Dynamically Changing the JService Trace Level .....	160
Resource Server Logging .....	160
Setting the Log Level .....	161
Name of the Log File .....	162
MapMaker Logging.....	162
Code Generation Log Files .....	163
MapPlayer Logging.....	163
Resource Database Server Logging .....	163
Configuration Manager Logging.....	163
System Monitor Logging.....	163
Service Logging .....	164
 <b>Chapter 9. National Language Support .....</b>	<b>165</b>
Single-Byte Character Sets .....	165
Double-byte Character Sets.....	166
 <b>Chapter 10. LAX Files .....</b>	<b>169</b>
Editing LAX Files .....	169
LAX File Content .....	169
Redirecting Standard Error and Standard Out.....	170
Changing the JVM Used to Launch JI Integration Applications .....	171
Example LAX File .....	171
 <b>Chapter 11. Licensing Procedures .....</b>	<b>177</b>
The License Key .....	178
License Properties .....	179
Feature Type .....	179
License Count .....	180
Expire Flag .....	180
The License File (license.txt) .....	181
Adding and Editing a License Key .....	182
Adding an Additional License Key Node.....	183
Adding an Additional License Data Source .....	184
Checking the Status of JI Integration Licenses .....	185
License Expiration Email Notification .....	187
Resource Server Configuration File ( <i>ressvr.cfg</i> ) Email Definitions .....	189
Duplicate Licenses .....	190
License Compatibility .....	191

<b>Chapter 12. Uninstalling JI Integration Software .....</b>	<b>193</b>
<b>Chapter 13. Field Attribute Definitions .....</b>	<b>195</b>
IBM 3270 Field Attribute Byte .....	195
IBM 5250 Field Attribute Byte .....	196
IBM 3270/5250 Interpreted Field Attribute Byte.....	197
<b>Glossary</b>	<b>199</b>
<b>Index .....</b>	<b>1</b>

---

## About this Guide

---

Welcome to the *JI Integration Supplemental Reference Guide*.

## Before You Begin

---

This guide is intended for software developers who want to create applications using JI Integration.

In order to use JI Integration, developers should have a working knowledge of the following:

- UNIX and/or Windows<sup>®</sup> system administration.
- Java<sup>™</sup> programming language for developing JI Integration services.
- The location and communication requirements of existing legacy data and applications.

Optional requirements include:

- Java programming languages for developing JI Integration clients.
- HTML design for using the JI Integration Common Gateway Interface.
- Siebel Enterprise Relationship Management applications to integrate JI Integration with Siebel environments.

## Organization

---

This guide is organized as follows:

- Chapter 1 - "JI Integration Commands" on page 9
- Chapter 5 - "JI Integration Error Messages" on page 87
- Chapter 6 - "Keyboard Mapping" on page 103
- Chapter 7 - "Configuration Files" on page 131
- Chapter 8 - "Logging Information" on page 151
- Chapter 9 - "National Language Support" on page 165
- Chapter 10 - "LAX Files" on page 169
- Chapter 11 - "Licensing Procedures" on page 177
- Chapter 12 - "Uninstalling JI Integration Software" on page 193
- Chapter 13 - "File Formats" on page 207
- Chapter 13 - "Field Attribute Definitions" on page 195

- “Glossary” on page 199
- “Index” on page 1

## Formatting Conventions

---

The following formatting conventions are used in this manual:

**Table 1. Formatting conventions**

Convention	Used for..
<i>Italics</i>	Italics are used for files, directories, programs, and book titles. For example: <code>&lt;JI_install_dir&gt;/bin/ea_mapmaker.exe</code> .
Monotype font	A monotype font is used to represent examples of code, characters that the user enters, and prompts or messages from the system. For example: Type <code>ea_start</code> at the command prompt.
<b>Sans-serif font</b>	A sans-serif font is used to represent Graphical User Interface (GUI) features, such as buttons. For example: Press the Help button to display a list of help topics.
<b>Serif bold font</b>	This font is used for notes and warnings that require special attention. For example: Warning: You must install JI Integration in an empty directory.

## Documentation Set

---

JI Integration is supplied with the manuals shown below. The documentation is delivered in Adobe Acrobat Reader Portable Document Format (PDF). No hardcopy documentation is provided, but you can print the PDF files on your local printer.

Use this guide in conjunction with other manuals provided with JI Integration:



**Table 2. JI Integration documentation set**

Title	Description
<i>JI Integration Release Notes</i>	Provides information about additions and revisions to the current release of JI Integration. The release notes are distributed in two forms, as a PDF and as a text file
<i>JI Integration Installation and Configuration Guide</i>	Details installation procedures for JI Integration.
<i>JI Integration Tutorial</i>	Provides hands-on instruction about how to write JI Integration services using MapMaker
<i>JI Integration User's Guide</i>	Describes how to configure the JI Integration environment, as well as how to use JI Integration graphical development and system monitoring tools.
<i>JI Integration Client Developer's Guide</i>	Describes how to design and develop JI Integration client applications, as well as how to integrate JI Integration services into third-party development environments.
<i>JI Integration Integration Guide</i>	Contains information about integrating JI Integration Services with other technologies, such as Siebel eBusiness Applications, MQSeries, Web Services, and more.
<i>JI Integration Supplemental Reference Guide</i>	Contains additional information on JI Integration commands, error codes, language translation, keyboard mapping, logging, licensing, file formats, and field attributes. This guide replaces the Appendices that were duplicated across the manual set.

## Viewing the Documentation Online

---

Online documentation is available in the following locations:

- In JI Integration Graphical User Interfaces (GUIs), click on the **Help** menu and select **Help Topics** to display the JI Integration documentation.
- JI Integration documentation is available on-line in Adobe® Acrobat™'s Portable Document Format (PDF). If the documentation has been installed, open the file `<JI_install_dir>/doc/_ji_doc.pdf` in Adobe Acrobat Reader™ or a Web browser (where `<JI_install_dir>` is the location of your JI Integration installation).

You can also access the latest version of the documentation for Software GmbH products at <http://documentation.softwareag.com/>. As new versions become available, the documentation on this web site will be updated and the previous versions will be migrated to the Empower Product Support Web site at <https://empower.softwareag.com/>. If you have a maintenance contract, you can view all versions of documentation on this web site. You will find instructions for registering and obtaining a userid and password on the documentation web site.

# Chapter 1. JI Integration Commands

---

This chapter describes the following JI Integration commands:

- “ea\_admin” on page 10
- “ea\_cfgmgr” on page 14
- “ea\_convertldb” on page 16
- “ea\_convertmap” on page 19
- “ea\_createdb” on page 20
- “ea\_dbping” on page 22
- “ea\_dbshutdown” on page 24
- “ea\_dbsvr” on page 26
- “ea\_envmgr” on page 29
- “ea\_exportdb” on page 32
- “ea\_importdb” on page 35
- “ea\_mapmaker” on page 38
- “ea\_mapplayer” on page 55
- “ea\_ping” on page 57
- “ea\_ressvr” on page 60
- “ea\_shutdown” on page 62
- “ea\_shutsrvr” on page 66
- “ea\_start” on page 68
- “ea\_status” on page 72
- “ea\_sysmon” on page 75
- “ji\_svc” on page 77

## ea\_admin

### Syntax

```
ea_admin <command> [<command_arg> ...]
  [-a | --aclAccount | --aclUser <ACL_account>]
  [-c | --eaEnv <JI_config_dir>] [-e | --extended] [-h | --host <host>]
  [-i | --jiInstallDir <JI_install_dir>] [-s | --envMgrPort <EnvMgr_port>]
  [-p | rmiPort <RMI_port>] [-x | --debug]
ea_admin -? | -help | --help
ea_admin -v | --version
```

### Parameters

#### Admin Command Parameters

<code>aclStatus, --aclStatus</code>	Print the status of JI Access Control Lists.
<code>checkACLUser &lt;User&gt; &lt;Pass&gt;, --checkACLUser &lt;User&gt; &lt;Pass&gt;</code>	Check the validity of a JI ACL user and password, where <User> and <Pass> are the ACL user and password..
<code>config [&lt;User&gt; &lt;Pass&gt;], --config [&lt;User&gt; &lt;Pass&gt;]</code>	Print the Environment Manager configuration, where <User> and <Pass> are the ACL user and password.
<code>environments, --environments</code>	Print a list of JI environments.
<code>jcluster &lt;Cluster_addr&gt;, --jcluster &lt;Cluster_addr&gt;</code>	Print the status for a JCluster, where <Cluster_addr> is the JCluster address.
<code>jclusters, --jclusters</code>	Print the status for all JClusters.
<code>licenseStatus, --licenseStatus</code>	Print the status of all JI licenses.

<code>plist [&lt;UserID&gt;],</code> <code>--plist [&lt;UserID&gt;]</code>	Print a list of JI processes. If the optional <userid> is specified, only processes for that user are listed.  <b>Note:</b> The plist command requires JI be installed with a Java Development Kit (JDK).
<code>resources, --resources</code>	Print a list of JI resources.
<code>status, --status</code>	Print the Environment Manager status.
<code>stop &lt;Cluster_addr&gt;,</code> <code>--stop &lt;Cluster_addr&gt;</code>	Stop the specified JCluster.

## JI Configuration Option Parameters

<code>-a &lt;ACL_account&gt;,</code> <code>--aclAccount &lt;ACL_account&gt;,</code> <code>--aclUser &lt;ACL_account&gt;</code>	Specify the ACL account name or user ID.
<code>-c &lt;JI_config_dir&gt;,</code> <code>--eaEnv &lt;JI_config_dir&gt;</code>	Location of the JI configuration files
<code>-i &lt;JI_install_dir&gt;,</code> <code>--jiInstallDir</code> <code>&lt;JI_install_dir&gt;</code>	JI installation directory
<code>-s &lt;EnvMgr_port&gt;,</code> <code>--envMgrPort &lt;EnvMgr_port&gt;</code>	Environment Manager listen port
<code>-p &lt;RMI_port&gt;,</code> <code>--rmiPort &lt;RMI_port&gt;</code>	RMI port for the JI Environment Manager and Resource Server.
<code>-h &lt;Host&gt;, --host &lt;Host&gt;</code>	Host address of the JI server.
<code>-w &lt;ACL_password&gt;,</code> <code>--aclPass &lt;ACL_password&gt;</code>	Specify the ACL password.

## Miscellaneous Option Parameters

-?, -help, --help	Print help information.
-e, --extended	Display extended information. This option applies to the "environments", "jclusters" and "jcluster" commands and has no effect on other commands.
-v, --version	Print version information.
-x, --debug	Print debug information.

## Description

The `ea_admin` command allows the user to perform the following administrative functions in the JI Integration environment:

- Check the status of the ACL system
- Check the validity of a ACL user name and password
- Print the environment manager configuration
- Print a list of JI Integration environments
- Print the environment manager status
- Print the status of all JClusters
- Print the detailed status of a JCluster
- Stop a JCluster
- Print the status of all JI Integration licenses
- Print a list of JI Integration processes
- Print a list of JI Integration resources

The `EA_ENV` environment variable must be set to the `<JI_install>JI_install_dir>/config` directory or the `-c <JI_config_dir>` command line option must be used. The `environment.ccf`, `ressvr.cfg`, and `envmgr.cfg` files must be included in this directory. The `environment.ccf` file must contain a definition of `EA_HOME` as well as the `DB_TCP_PORT` parameter, which is required to locate the Resource Database server. If the `-i <JI_install_dir>` command line flag is used, `EA_HOME` does not have to be set. Also, `EA_HOME` and `DB_TCP_PORT` can be defined as environment variables if the `environment.ccf` file is not used.

In order to execute the `ea_admin` command correctly, this program must be able to determine the port numbers of the server components and the location of the appropriate `java` command to use to execute the underlying java executables that shut down the various components of the environment.

The program determines this information as follows:

- **Port Numbers of the Server Components.** The port numbers for each of the server components are determined as follows (in order of precedence):
  - Environment Manager  
Command line option (-s <EnvMgr\_port>)  
The value of the EA\_ENVMGR parameters in *environment.ccf*.  
The value of the EA\_ENVMGR environment variable.  
The default value for the Environment Manager port (30001)
  - Resource Server  
Command line option (-p <RMI\_port>)  
The value of the config.rmiport parameter in *ressvr.cfg*, the default Resource Server configuration file. The location of the *ressvr.cfg* file is assumed to be the <JI\_install\_dir>/config directory.  
The default value for the Resource Server port (30002).
  - Resource Database Server  
The value of the DB\_TCP\_PORT parameter in *environment.ccf*.  
The value of the DB\_TCP\_PORT environment variable.  
The default Resource Database server listen port (30000)
- **Java command.** Because *ea\_admin* executes underlying Java programs to shut down the JI Integration environment, it must be able to locate the correct Java Virtual Machine (JVM) executable (*java* in UNIX or *java.exe* in Windows). The location of the JVM is located by using either:
  - The value of the javacmd parameter in *envmgr.cfg*, the default Environment Manager configuration file. The location of the *envmgr.cfg* file is assumed to be the <JI\_install\_dir>/config directory.
  - If the *envmgr.cfg* file is not found or the javacmd parameter is not set, then *java* will be used as a default command line. If this option is used, the JVM executable must be in the PATH environment variable.

## See Also

ea\_ping  
ea\_status

## ea\_cfgmgr

### Syntax

```
ea_cfgmgr [-f config_file] [-i install_dir] [-#]  
ea_cfgmgr -?, -h, -H  
ea_cfgmgr -v, -V
```

### Parameters

-f config_file	Identifies the name of the configuration file for the Configuration Manager. If this parameter is not used, the default value, <i>cfgmgr.cfg</i> , is used. Optionally, the path can be identified in this parameter as well. If no path is identified, the default value, <i>&lt;JI_install_dir&gt;/config</i> is used.
-i install_dir	Identifies the path to the root directory of the installation of JI Integration.
-#	Determines the level of debug logging. Allowable values for # are 0 through 9, with 0 representing minimal logging and 9 representing the most verbose logging.
-, -h, -H	Causes help information to be written to the <i>cfgmgr_stdout.txt</i> file, located in <i>&lt;JI_install_dir&gt;/logs</i> .
-v, -V	Causes version information about the Configuration Manager to be written to the <i>cfgmgr_stdout.txt</i> file, located in <i>&lt;JI_install_dir&gt;/logs</i> .

### Description

Launches the JI Integration Configuration Manager, used to configure resources in your JI Integration environment. A configuration file is used to identify configuration parameters for the Configuration Manager. The default name of the configuration file, *cfgmgr.cfg*, can be overridden using the `-f config_file` command line option when the Configuration Manager is launched. This option can also be used to indicate the path to the configuration file. If no path is



identified, the default path `<JI_install_dir>/config` is used. All command line options override the equivalent settings in the Configuration Manager configuration file.

### **See Also**

ea\_mapmaker  
ea\_sysmon

## ea\_convertldb

### Syntax

```
ea_convertldb [-b | --dbHome <dir>] [-D | --debug]
               [-d | --jiInstallDir <dir>] [-H | --srcHost <host>] [-n | --srcName <dbname>]
               [-P | --srcPass <pass>] [-p | --srcPort <port>] [-S | --srcSocket <sock>]
               [-u | --srcUser <user>] [-v | --verbose] output-file
ea_convertldb -? | -h | -help | --help
ea_convertldb -V | --version
```

### Parameters

<code>-b &lt;dir&gt;, --dbHome &lt;dir&gt;</code>	Specify the source database installation directory.
<code>-D, --debug</code>	Enable debug logging.
<code>-d &lt;dir&gt;, --jiInstallDir &lt;dir&gt;</code>	Specify the JI installation directory. The JI installation directory is used to determine where to create temporary files.
<code>-, -h, -help, --help</code>	Print the help message.
<code>-H &lt;host&gt;, --srcHost &lt;host&gt;</code>	Specify the source database host name or address. The default is: "localhost".
<code>-n &lt;dbname&gt;, --srcName &lt;dbname&gt;</code>	Specify the name of the source database to convert. The default is: "access").
<code>-P &lt;pass&gt;, --srcPassword &lt;pass&gt;</code>	Specify the source database password.
<code>-p &lt;port&gt;, --srcPort &lt;port&gt;</code>	Specify the source database port. The default is: 30000.
<code>-S &lt;sock&gt;, --srcSocket &lt;sock&gt;</code>	Specify the source database server UNIX domain socket. The default is: "<JI_install_dir>/databases/mysql.sock").

<code>-u &lt;user&gt;, --srcUser &lt;user&gt;</code>	Specify the source database user id.
<code>-V, --version</code>	Print the program version and exit.
<code>-v, --verbose</code>	Enable verbose logging.
<code>output-file</code>	The name of the output file (the converted database dump) to be created. This option can also identify the path to the file; if no path is identified, the file is created in the same directory from which the <code>ea_convertddb</code> command was launched. This parameter is required.

## Description

The `ea_convertddb` command may be used to convert a MySQL (JI 4.5.5 and earlier) Resource Database to an H2 Database Engine compatible export format which can then be imported into an H2 (JI 4.5.6 and newer) Resource Database.

The `output-file` parameter is required. Unless an absolute path is specified for the output file, the output file location will be relative to the the directory from which `ea_convertddb` was started. The `ea_convertddb` command uses the following default parameter values:

- source host = localhost
- source port = 30000
- source db name = access

The `ea_convertddb` command must be run from the target JI environment. The source JI Resource Database server must be running in order perform the conversion process.

## Examples

In the following examples, `<trgt-env-dir>` is the path to the target JI Integration installation on the local machine. It is assumed that the source Resource Database server is running and that it is listening on `<src-port>` on the machine which the source environment is running on.

### Example 1:

The following command:

```
<trgt-env-dir>/bin/ea_convertldb access.exp
```

will connect to the source Resource Database server listening on the default port (port 30000) on the local machine and iterate through the tables in the database named *access*, converting each table to H2 export format and write the results to *access.exp*.

### Example 2:

The following command:

```
<trgt-env-dir>/bin/ea_convertldb -H <src-host> -p <src-port>  
-n <ji-dbname> <output-file>
```

will connect to the remote source Resource Database server running on <src-host> listening on <src-port> and iterate through the tables in the <ji-dbname> database converting each table to H2 export format and write the results to <output-file>.

### Example 3:

This example does the same thing as "Example 2:"using the long command line options:

```
<trgt-env-dir>/bin/ea_convertldb --srcHost <src-host> --srcPort <src-port>  
--srcName <ji-dbname> <output-file>
```

## See Also

ea\_createdb

ea\_exportdb

ea\_importdb

## ea\_convertmap

### Syntax

```
ea_convertmap [-s | --noStructName] mapfile
ea_createdb -? | -h | -help | --help
ea_createdb -V | --version
```

### Parameters

-?, -h, -help, --help	Display a help message.
-s	Do not set Structure Name on converted OnFail Invoke steps.
-V, --version	Show the program version.
mapfile	Map file to convert.

### Description

Launches the JI Integration map conversion program, which is used to convert JI version 3.5 maps for use with JI 4.x. All JI 3.5 maps must be converted before they can be loaded in JI 4.x MapMaker. A backup of the original map named <mapfile>.old will be created during the conversion process.

For example, the following command:

```
ea_convertmap myService.map
```

will convert the myService.map from JI 3.5 format to JI 4.x format, leaving a backup of the original map in myService.map.old.

### See Also

ea\_mapmaker

## ea\_createdb

### Syntax

```
ea_createdb [-b | --dbHome <dir>] [-D | --debug]
            [-d | --jiInstallDir <dir>] [-H | --dbHost <host>]
            [-n | --dbName <dbName>] [-p | --dbPort <port>] [-S | --dbSock socket]
            [-u | --dbUser <user>] [-v | --verbose]
ea_createdb -? | -h | -help | --help
ea_createdb -V | --version
```

### Parameters

<code>-b &lt;dir&gt;, --dbHome &lt;dir&gt;</code>	Database Installation directory.
<code>-D, --debug</code>	Enable debug logging.
<code>-d &lt;dir&gt;, --jiInstallDir &lt;dir&gt;</code>	Identifies the path to the root directory of an installation of JI Integration. The default is <i>&lt;JI_install_dir&gt;</i> (the directory in which JI Integration was installed).
<code>-H &lt;host&gt;, --dbHost &lt;host&gt;</code>	The host machine on which the Resource Database server is running. This option defaults to localhost.
<code>-?, -h, -help, --help</code>	Print the help message.
<code>-n &lt;dbname&gt;, --dbName &lt;dbname&gt;</code>	Name of database to create (default = "access").
<code>-P &lt;pass&gt;, --dbPass &lt;pass&gt;</code>	The database password.
<code>-p port, --dbPort &lt;port&gt;</code>	Identifies the port on which the database daemon is running. The default setting for <i>&lt;port&gt;</i> is 30000.
<code>-S socket, --dbSock &lt;sock&gt;</code>	The UNIX domain socket on which the Resource Database server is running.

<code>-u &lt;user&gt;, --dbUser &lt;user&gt;</code>	The database user ID.
<code>-V, --version</code>	Causes version information about the Create Database program to be written to the <i>createdb_stdout.txt</i> file, located in <i>&lt;JI_install_dir&gt;/logs</i> .
<code>-v, --verbose</code>	Causes verbose output to be to be written to the <i>createdb_stdout.txt</i> file, located in <i>&lt;JI_install_dir&gt;/logs</i>

**Note:** On UNIX, output will go to stdout, not to the *createdb\_stdout.txt* file.

## Description

Launches the JI Integration Create Database program, which is used to create new Resource Databases. For more information about Resource Databases, see the *JI Integration User's Guide*.

When no parameters are specified for the Create Database program, it will create a database using the following default settings:

- `dbname = access`
- `host = localhost`
- `port = 30000`
- `dir = <JI_install_dir>` (the directory in which JI Integration was installed).

For example, the following command:

```
ea_createdb -n test_db -p 14524 -d /usr/local/EnterpriseAccess
```

will create a database named “test\_db” using the ResourceDatabase server running on the local host, port 14524. The database will be created in the database directory being used by that instance of the Resource Database server.

## See Also

`ea_exportdb`  
`ea_importdb`

## ea\_dbping

### Syntax

```
ea_dbping [-H | --dbHome <dir>] [-h | --dbHost <host>]
          [-i | --jiInstallDir <dir>] [-P | --dbPass <dbPass>]
          [-p | --dbPort <port>] [-u | --dbUser <user>] [-v | --verbose]
          [-x | --debug]
ea_dbping -? | -h | -help | --help
ea_dbping -V | --version
```

### Parameters

<code>-H &lt;dir&gt;, --dbHome &lt;dir&gt;</code>	Database Installation directory.
<code>-h &lt;host&gt;, --dbHost &lt;host&gt;</code>	The host machine on which the JI DB server is running. This option defaults to localhost.
<code>-?, -help, --help</code>	Print the help message.
<code>-i &lt;dir&gt;, --jiInstallDir &lt;dir&gt;</code>	Identifies the path to the root directory of an installation of JI Integration. The default is <i>&lt;JI_install_dir&gt;</i> (the directory in which JI Integration was installed).
<code>-P &lt;pass&gt;, --dbPass &lt;pass&gt;</code>	The JI DB Server password.
<code>-p &lt;port&gt;, --dbPort &lt;port&gt;</code>	Identifies the port on which the JI DB server is running. The default setting for <i>&lt;port&gt;</i> is 30000.
<code>-u &lt;user&gt;, --dbUser &lt;user&gt;</code>	The JI DB server user ID.
<code>-V, --version</code>	Print the program version information and exit.
<code>-v, --verbose</code>	Enable verbose logging.
<code>-x, --debug</code>	Enable debug logging.



## Description

The `ea_dbping` command is used by the `ea_ping` command to determine if the JI Integration Resource Database server is running. For more information about Resource Databases, see the *JI Integration User's Guide*.

When no parameters are specified, `ea_dbping` will use the following default settings:

- `host = localhost`
- `port = 30000`

For example, the following command:

```
ea_dbping -p 14524 -h prdserve
```

will attempt to ping the ResourceDatabase server running on `prdserve`, port 14524.

**Note:** The `ea_ping` command should be used for normal operations.

## Exit Status

If the server is alive, `ea_dbping` will exit with a status of 0. If the server is not running or there is an error, `ea_dbping` will exit with a non-zero status.

## See Also

`ea_dbshutdown`  
`ea_dbsvr`  
`ea_ping`

## ea\_dbshutdown

### Syntax

```
ea_dbshutdown [-a | --all] [-f | --force] [-h | --dbHost <host>]
               [-i | --jiInstallDir <dir>] [-P | --dbPass <dbPass>]
               [-p | --dbPort <port>] [-u | --dbUser <user>] [-v | --verbose]
               [-x | --debug]
ea_dbshutdown -? | -help | --help
ea_dbshutdown -V | --version
```

### Parameters

-a, --all	Shutdown all JI db servers.
-f, --force	Force the shutdown of the JI DB server.
-h <host>, --dbHost <host>	The host machine on which the JI DB server is running. This option defaults to localhost.
-?, -h, -help, --help	Print the help message.
-i <dir>, --jiInstallDir <dir>	Identifies the path to the root directory of an installation of JI Integration. The default is <i>&lt;JI_install_dir&gt;</i> (the directory in which JI Integration was installed).
-P <pass>, --dbPass <pass>	The JI DB server password.
-p <port>, --dbPort <port>	Identifies the port on which the JI DB server is running. The default setting for <i>&lt;port&gt;</i> is 30000.
-u <user>, --dbUser <user>	The JI DB server user ID.
-V, --version	Print the program version information and exit.
-v, --verbose	Enable verbose logging.

-x, --debug                      Enable debug logging.

## Description

The `ea_dbshutdown` program is used by the `ea_shutdown` command to shutdown the JI Integration Resource Database server. For more information about Resource Databases, see the *JI Integration User's Guide*.

When no parameters are specified, `ea_dbshutdown` will use the following default settings:

- `host = localhost`
- `port = 30000`

For example, the following command:

```
ea_dbshutdown -p 14524 -h prdserve
```

will attempt to shutdown the ResourceDatabase server running on `prdserve`, port 14524.

**Note:** The `ea_shutdown` command should be used for normal environment shutdown.

## Exit Status

If the JI DB server is shut down successfully, `ea_dbshutdown` will exit with a status of 0. If the server is not running or there is an error shutting down the server, `ea_dbshutdown` will exit with a non-zero status.

## See Also

`ea_dbping`  
`ea_dbsvr`  
`ea_shutdown`

## ea\_dbsvr

### Syntax

```
ea_dbsvr [-a | --dbBase <dir>] [-B | --dbHome <dir>] [--daemon]
  [-h | --dbHost <host>] [-i | --jiInstallDir <jiInstallDir>]
  [-l | dbLog <logPath>] [-n | --retries <pingRetries>]
  [-P | --dbPass <dbPass>] [-p | --dbPort <port>] [-t | --sleepTime <ms>]
  [-U | --dbSock <dbSockPath>] [-u | --dbUser <dbUser>] [-V | --version]
  [-v | --verbose] [-x | --debug]
ea_dbsvr -? | -help --help
ea_dbsvr -V | --version
```

### Parameters

<code>-a &lt;dir&gt;, --dbBase &lt;dir&gt;</code>	Identifies the path to the JI DB server data directory. The default is <code>&lt;JI_install_dir&gt;/databases</code> .
<code>-B &lt;dir&gt;, --dbHome &lt;dir&gt;</code>	Identifies the path to the JI DB server installation directory. The default is <code>&lt;JI_install_dir&gt;</code> (the directory in which JI Integration was installed).
<code>--daemon</code>	Start the JI DB server as a daemon or service process.
<code>-h &lt;host&gt;, --dbHost &lt;host&gt;</code>	The host machine on which the JI DB server is running. This option defaults to <code>localhost</code> .
<code>-?, -h, -help, --help</code>	Print the help message.
<code>-i &lt;dir&gt;, --jiInstallDir &lt;dir&gt;</code>	Identifies the path to the root directory of an installation of JI Integration. The default is <code>&lt;JI_install_dir&gt;</code> (the directory in which JI Integration was installed).
<code>-l &lt;logPath&gt;, --dbLog &lt;logPath&gt;</code>	Identifies the path to the JI DB server log file. The default is <code>&lt;JI_install_dir&gt;/logs/dbsvr-&lt;port&gt;.log</code>

<code>-n &lt;ms&gt;, --retries &lt;ms&gt;</code>	When used in conjunction with the <code>--daemon</code> option, the number of times to ping the JI DB server while waiting for the daemon to start. The default is 10.
<code>-P &lt;pass&gt;, --dbPass &lt;pass&gt;</code>	The JI DB server password.
<code>-p &lt;port&gt;, --dbPort &lt;port&gt;</code>	Identifies the port on which the JI DB server is running. The default setting for <code>&lt;port&gt;</code> is 30000.
<code>-t &lt;ms&gt;, --sleepTime &lt;ms&gt;</code>	When used in conjunction with the <code>--daemon</code> option, the amount of time, in milliseconds to sleep between ping retries while waiting for the daemon to start. The default is 500.
<code>-U &lt;sockPath&gt;, --dbSock &lt;sockPath&gt;</code>	The path to the UNIX socket on which the JI DB server is listening.
<code>-u &lt;user&gt;, --dbUser &lt;user&gt;</code>	The JI DB server user ID.
<code>-V, --version</code>	Print the program version information and exit.
<code>-v, --verbose</code>	Enable verbose logging.
<code>-x, --debug</code>	Enable debug logging.

## Description

The `ea_dbsvr` command is used by the `ea_start` command to start the JI Integration Resource Database server. For more information about Resource Databases, see the *JI Integration User's Guide*.

When no parameters are specified, `ea_dbsvr` will use the following default settings:

- `port = 30000`

For example, the following command:

```
ea_dbsvr -p 14524
```

will attempt to start the JI ResourceDatabase server running on port 14524.

**Note:** The `ea_start` command should be used for normal environment startup.

## Exit Status

If the `ea_dbsvr` command is run with the `--daemon` option, it will attempt to start the JI DB server in the background and will wait for the server to start. If the JI DB server is started successfully, `ea_dbsvr` will exit with a status of 0. If the server is not started or there is an error, `ea_dbsvr` will exit with a non-zero status.

If the `ea_dbsvr` command is run without the `--daemon` option, it will start the server and wait for the server to be shut down. The `ea_dbsvr` command will then exit with a status of 0 if there were no errors or a non-zero status if there were any errors.

## See Also

`ea_dbping`  
`ea_dbshutdown`  
`ea_start`

## ea\_envmgr

### Syntax

```
ea_envmgr [-a <aclRes>] [-d <rmiPortNum>] [-f <cfgFile>] [-g <ldBalGrp>]
          [-i <installDir>] [-l <logFile>] [-#] [-p <svrPortNum>] [-r <cfgRes>]
ea_envmgr -? | -h | -H | -help | --help
ea_envmgr -v | -V | --v | --V
```

### Parameters

-a <aclRes>	Identifies the name of the Access Control List resource.
-d rmiport	Identifies the port for Remote Method Invoke (RMI) activity for the Environment Manager and the Resource Server. If the Resource Server and Environment Manager are running on the same machine, the value of this port can match the value for the config.rmiport parameter in the Resource Server configuration file or the -p rmiport command line option for the Resource Server. However, if the same RMI port is used for the Environment Manager and the Resource Server and the Resource Server goes down, the Environment Manager will have to be restarted. This option will override the rmiport record in the Resource Database for the Environment Manager's configuration.
-f config_file	Identifies the name of the configuration file for the Environment Manager. If this parameter is not used, the default value, evnmgr.cfg, is used. Optionally, the path can be identified in this parameter as well. If no path is identified, the path identified in the install_dir parameter is used.
-g groupname	Identifies the name of the Environment Manager's load balancing group. This option will override the value of the groupname parameter in the Environment Manager's configuration file, and will also override the group record in the Resource Database for the Environment Manager's configuration.

<code>-?, -h, -H, --h, --H</code>	Causes help information to be written to the <code>envmgr_stdout.txt</code> file, located in <code>&lt;JI_install_dir&gt;/logs</code> .
<code>-i install_dir</code>	Identifies the path to the installation of JI Integration.
<code>-l logfile</code>	Identifies the name of the logfile for the Environment Manager. Optionally, the path can be identified in this parameter as well. If no path is identified, the default value <code>&lt;JI_install_dir&gt;/logs</code> is used.
<code>-#</code>	Determines the level of debug logging. Allowable values for # are 0 through 9, with 0 representing minimal logging and 9 representing the most verbose logging.
<code>-p serverport</code>	Identifies the port on which the Environment Manager will listen for client connections. When the client identifies the host and port of the Environment Manager, this is the port that should be identified. This option will override the value of the <code>serverport</code> parameter in the Environment Manager's configuration file, and will also override the <code>serverport</code> record in the Resource Database for the Environment Manager's configuration.
<code>-r resourcename</code>	Identifies the name of the Resource Database that contains configuration information for the Environment Manager. The location of the database is defined in the Resource Server node in the Configuration Manager; therefore, the value of this parameter must correspond to the name of the appropriate resource listed in the Configuration Manager. This option will override the value of the <code>resourcename</code> parameter in the Environment Manager's configuration file.
<code>-s</code>	Causes the Environment Manager to run as a Windows service (Windows only).
<code>-v, -V, --v, --V</code>	Causes version information about the Environment Manager to be written to the <code>envmgr_stdout.txt</code> file, located in <code>&lt;JI_install_dir&gt;/logs</code> .



**Note:** Only the first character of the command line parameter is significant. For example, “-version” is equivalent to “-v”.

## Description

Launches the JI Integration Environment Manager. Environment Managers serve as the central management, access, and control points for JI Integration server-side processing. They are used to manage JClusters, JServices, and to provide multicasting information to locate resources and balance load.

A configuration file is used to identify configuration parameters for the Environment Manager. The default name of the configuration file, *envmgr.cfg*, can be overridden using the `-f config_file` command line option when the Environment Manager is launched. This option can also be used to indicate the path to the configuration file; if no path is identified, the default path `<JI_install_dir>/config` is used. All command line options override the equivalent settings in the Environment Manager configuration file.

## See Also

`ea_ressvr`

## ea\_exportdb

### Syntax

```
ea_exportdb [-b | --dbHome <dir>] [-D | --debug] [-d | --jiInstallDir <dir>]
  [-H | --dbHost <host>] [-n | --dbName <dbname>] [-P | --dbPass <pass>]
  [-p | --dbPort <port>] [-S | --dbSocket <sock>] [-u | --dbUser <user>]
  [-v | --verbose] [tables] output-file
ea_exportdb -? | -h | --h | -help | --help
ea_exportdb -V | --version
```

### Parameters

<code>-b &lt;dir&gt;, --dbHome &lt;dir&gt;</code>	Specify the database base installation directory. The default is: <i>&lt;JI_install_dir&gt;</i> .
<code>-D, --debug</code>	Enable debug logging.
<code>-d &lt;dir&gt;, --jiInstallDir &lt;dir&gt;</code>	Specify the path to the root directory of an installation of JI Integration.
<code>-H &lt;host&gt;, --dbHost &lt;host&gt;</code>	Specify the host machine on which the Resource Database server is running. This option defaults to localhost.
<code>-, -h, -help, --help</code>	Write the help message to the <code>exportdb_stdout.txt</code> file, located in <i>&lt;JI_install_dir&gt;/logs</i> .
<code>-n &lt;dbname&gt;, --dbName &lt;dbname&gt;</code>	The name of the database which will be exported. The default is: "access".
<code>-P &lt;pass&gt;, --dbPass</code>	Database password.
<code>-p &lt;port&gt;, --dbPort &lt;port&gt;</code>	Specify the port on which the Resource Database server is running. The default setting for port is 30000.
<code>-S &lt;sock&gt;, --srcSocket &lt;sock&gt;</code>	The UNIX domain socket on which the Resource Database server is running. Deprecated.

<code>-u &lt;user&gt;, --dbUser &lt;user&gt;</code>	Database user ID.
<code>-V, --version</code>	Causes version information about the Export Database program to be written to the <code>exportdb_stdout.txt</code> file, located in <code>&lt;JI_install_dir&gt;/logs</code> .
<code>-v, --verbose</code>	Enable verbose logging to the <code>exportdb_stdout.txt</code> file, located in <code>&lt;JI_install_dir&gt;/logs</code> .
<code>tables</code>	Names of the tables to export. This option can not be used in conjunction with the <code>-l</code> or <code>-s</code> options.
<code>output-file</code>	The name of the output file (the database dump) file to be created. This option can also identify the path to the file; if no path is identified, the file is created in the same directory from which the Export Database file was launched. This option is required.

**Note:** On UNIX, output will go to stdout, not to the `exportdb_stdout.txt` file.

## Description

Launches the JI Integration Export Database program, which is used to export data from the Resource Database into a file. This file can then be imported into another database using the Import Database program (`ea_importdb`). For more information about Resource Databases, see the *JI Integration User's Guide*.

The `-d install_dir` and `output-file` parameters are required. When no other parameters are specified, the Export Database program will export the database using the following default settings:

- `dbname` = `access`
- `host` = `localhost`
- `port` = `30000`

For example, the following command: `ea_exportdb -n test_db -p 14524 -d /usr/local/EnterpriseAccess /tmp/test_db.dump` will create a file named `"test_db.dump"` based on the data in the database named `"test_db,"` using the Resource Database server running on the local host, port 14524. The export file will be created in the `/tmp` directory.

## See Also

`ea_createdb`  
`ea_importdb`

## ea\_importdb

### Syntax

```
ea_importdb [-b | --dbHome <dir>] [-D | --debug] [-d | --jiInstallDir <dir>]
             [-H | --dbHost <host>] [-n | --dbName <dbname>] [-P | --dbPass <pass>]
             [-p | --dbPort <port>] [-S | --dbSock <socket>] [-u | --dbUser <user>]
             [-v | --verbose] input-file
ea_importdb -? | -h | -help | --help
ea_importdb -V | --version
```

### Parameters

-b <dir>, --dbHome <dir>	Specify the database base installation directory. The default is <JI_install_dir>.
-D, --debug	Enable debug logging.
-d <dir>, --jiInstallDir	Specify the path to the root directory of an installation of JI Integration. This parameter is required.
-H <host>, --dbHost <host>	The host machine on which the Resource Database server is running. This option defaults to localhost.
-, -h, -help, --help	Write the help message to the importdb_stdout.txt file, located in <JI_install_dir>/logs.
-n <dbname>, --dbName <dbname>	The name of the database into which this data will be imported. The default is access.
-P <pass>, --dbPass <pass>	Database password.
-p <port>, --dbPort <port>	Specify the port on which the Resource Database server is running. The default setting for port is 30000.
-S <socket>, --dbSock <socket>	The UNIX domain socket on which the Resource Database server is running. Deprecated.

<code>-V, --version</code>	Write version information about the Import Database program to the <code>importdb_stdout.txt</code> file, located in <code>&lt;JI_install_dir&gt;/logs</code> .
<code>-u &lt;user&gt;, --dbUser &lt;user&gt;</code>	Database user id
<code>-v, --verbose</code>	Enable verbose logging to the <code>importdb_stdout.txt</code> file, located in <code>&lt;JI_install_dir&gt;/logs</code> .
<code>input-file</code>	The name of the input file (the database dump) file from which information will be imported. In Windows, the fully qualified path name is required. This option is required.

**Note:** On UNIX, output will go to stdout, not to the `importdb_stdout.txt` file.

## Description

Launches the JI Integration Import Database program, which is used to import data from a file into the Resource Database. The file that contains the import data is created when the Export Database program (`ea_importdb`) is used to export data from a database. For more information about Resource Databases, see the *JI Integration User's Guide*.

The `-d install_dir` and `input-file` parameters are required. When no other parameters are specified, the Export Database program will export the database using the following default settings:

- `dbname = access`
- `host = localhost`
- `port = 30000`

For example, the following command:

```
ea_importdb -n new_db -p 14524 -d /usr/local/EnterpriseAccess /tmp/  
test_db.dump
```

will import data from the file named “`test_db.dump`” in the `/tmp` directory. The data will be imported into the database named “`new_db`,” using the Resource Database server running on the local host, port 14524.

## **See Also**

ea\_createdb  
ea\_exportdb

## ea\_mapmaker

### Syntax

```
ea_mapmaker [-debug | --debug] [-f <cfgFile>] [-font <fontName>]
             [-s | --noStructName] [-verbose | --verbose] [map_file]
ea_mapmaker [-debug | --debug] [-verbose | --verbose] -batch <cmdFile>
ea_mapmaker [-debug | --debug] [-verbose] -showfonts
ea_mapmaker -? | -h | -help | --help
ea_mapmaker -v | -version | --version
```

**Note:** Any files passed as arguments on the MapMaker command line must be specified relative to the JI installation directory or as absolute paths.

### Parameters

Parameter	Description
-batch, --batch <cmdFile>	Perform the commands described in <cmdFile>.
-debug, --debug	Enable debug logging.
-f <cfgFile>	Use the specified configuration file.
-font, --font <fontName>	Use the specified font in the Emulator panel.
-, -h, -help --help	Causes help information to be written to the mapmaker_stdout.txt file, located in <JI_install_dir>/logs.
-s, --noStructName	Do not set Structure Name on new OnFail Invoke steps.
-showfonts, --showfonts	List the fonts available to Mapmaker in the stdout log file.



<code>-v, -version,</code> <code>--version</code>	Causes version information about MapMaker to be written to the <code>mapmaker_stdout.txt</code> file, located in <code>&lt;JI_install_dir&gt;/logs</code> .
<code>-verbose,</code> <code>--verbose</code>	Enable verbose logging.
<code>map_file</code>	Specifies the path and filename of the map file. Must have a <code>.map</code> extension.

**Note:** Command line options are case insensitive.

## Description

Launches the JI Integration MapMaker program, used to generate JI Integration Java services and test clients that interact with legacy applications. For more information about developing clients and services with MapMaker, see the *JI Integration User's Guide*.

## Command Line Batch Mode

The `-batch` command line option allows the user to script or automate tasks that would normally be performed interactively through the MapMaker graphical user interface.

**Tasks** The following tasks may be performed using the `-batch` command line option:

- Generation of JI clients
- Generation of JI services
- Generation of JI service definitions
- Deployment of JI Java services
- Listing deployed JI web services
- Deploying JI web services
- Undeploying JI web services

**Note:** It is not currently possible to generate a JI Resource Adapter with the MapMaker batch processing facility.

Depending on the batch properties file, the tasks are performed in the following order:

- 1 MapMaker checks for a valid license.
- 2 The specified map is loaded.
- 3 The batch properties file is loaded.
- 4 The map is updated.
- 5 The JI service is generated.
- 6 The service interface definitions (XML, DTD, XML Schema and WSDL) are generated.
- 7 The JI clients (JClient3, JSP and ASP.Net) are generated.
- 8 Deployed web services are listed.
- 9 Web services are undeployed.
- 10 Web services are deployed.
- 11 The JI service is deployed.

Only those tasks specified by the batch properties file are performed during the batch run.

### **Batch Processing Properties**

The tasks performed during JI batch processing are controlled by the command or properties file specified as an argument to the `-batch` command line option. The following tables list the batch processing properties:

## Generation Action Properties

Property Name	Description
<code>action.generate.service</code>	Boolean - Enable service generation if true. Default: <code>false</code> .
<code>action.generate.client</code>	Boolean - Enable client generation if true. Default: <code>false</code> .
<code>action.generate.definitions</code>	Boolean - Enable service definition generation if true. Default: <code>false</code> .

## Service Deployment Action properties

Property Name	Description
<code>action.deploy.jiSvc</code>	Boolean - Enable deployment of the generated JI service JAR to the JI ResSvr. Default: <code>false</code> .
<code>action.deploy.webSvc</code> <code>.listDeployedSvcs</code>	Boolean - List the name and URL of web services deployed to the application server. Default: <code>false</code> .
<code>action.deploy.webSvc.deploySvc</code>	Boolean - Enable deployment of the specified JI web service to the application server. Default: <code>false</code> .
<code>action.deploy.webSvc.undeploySvc</code>	Boolean - Enable undeployment of the specified JI web service from the application server. Default: <code>false</code> .

## General Action Properties

Property Name	Description
<code>action.updateMap</code>	Boolean - Enable update of the loaded map. Default: <code>false</code> .
<code>action.debug.enable</code>	Boolean - Enable debug logging. Default: <code>false</code> . <b>Note:</b> Debug logging may also be enabled via the <code>-debug</code> command line switch.
<code>action.verbose.enable</code>	Boolean - Enable verbose logging. Default: <code>false</code> . <b>Note:</b> Verbose logging may also be enabled via the <code>-verbose</code> command line switch.
<code>ji.installDir</code>	String path - The path to the JI installation. Default: unset. Overridden by the value from the MapMaker config file, the value in the map and the value in the batch properties file.

## Generation Properties

Property Name	Description
<code>gen.basePath</code>	String path - The base path for generated code. Default: unset. Overridden by the value from the MapMaker config file, the value in the map and the value in the batch properties file.

**Property Name****Description**`gen.svc.packageName`

String - Service Package Name

Default: unset. Overridden by the value from the MapMaker config file, the value in the map and the value in the batch properties file.

**Service Generation Properties****Property Name****Description**`gen.svc.supplementalImportPackages`

String - Colon or semi-colon separated list of additional packages that will be imported into the generated service.

Default: unset. Overridden by the value from the map and the value in the batch properties file.

`gen.svc.overWriteCustomCode`

Boolean - Re-generate and overwrite custom code if true.

Default: `false`. Overridden by the value from the map and the value in the batch properties file.

**Client Generation Properties****Property Name****Description**`gen.client.html`

Boolean - Generate HTML sample code for the service.

Default: `false`. Overridden by the value from the map and the value in the batch properties file.

Property Name	Description
<code>gen.client.jclient</code>	<p>Boolean - Generate JClient sample code for the service.</p> <p>Default: <code>false</code>. Overridden by the value from the map and the value in the batch properties file.</p>
<code>gen.client.jclient3</code>	<p>Boolean - Generate JClient3 sample code for the service.</p> <p>Default: <code>false</code>. Overridden by the value from the map and the value in the batch properties file.</p>
<code>gen.client.JSP</code>	<p>Boolean - Generate JSP sample code for the service.</p> <p>Default: <code>false</code>. Overridden by the value from the map and the value in the batch properties file.</p>
<code>gen.client.TSC</code>	<p>Boolean - Generate TSC sample code for the service.</p> <p>Default: <code>false</code>. Overridden by the value from the map and the value in the batch properties file.</p>
<code>gen.client.ASP.Net</code>	<p>Boolean - Generate ASP.Net sample code for the service.</p> <p>Default: <code>false</code>. Overridden by the value from the map and the value in the batch properties file.</p>
<code>gen.client.envMgr</code>	<p>String - EnvMgr host and port (host:port).</p> <p>Default: <code>unset</code>. Overridden by the value from the MapMaker config file, the value in the map and the value in the batch properties file.</p>

**Property Name****Description**`gen.client.sessionName`

String - JI client/service session name.

Default: unset. Overridden by the value from the map and the value in the batch properties file.

`gen.client.pathTo.NetWSDL.exe`

String - The path to the .Net framework SDK wsdl.exe program.

Default: unset. Overridden by the value from the map and the value in the batch properties file.

`gen.client.pathTo.NetCSC.exe`

String - The path to the .Net framework SDK csc.exe program.

Default: unset. Overridden by the value from the map and the value in the batch properties file.

`gen.client.includeSampleInputData`

Boolean - Include sample input data if true.

Default: unset. Overridden by the value from the map and the value in the batch properties file.

**Service Definition Generation Properties****Property Name****Description**`gen.definitions.XML`

Boolean - Enable generation of XML definitions.

Default: unset. Overridden by the value from the map and the value in the batch properties file.

Property Name	Description
<code>gen.definitions.DTD</code>	<p>Boolean - Enable generation of DTD definitions.</p> <p>Default: unset. Overridden by the value from the map and the value in the batch properties file.</p>
<code>gen.definitions.XMLSchema</code>	<p>Boolean - Enable generation of XML schema definitions.</p> <p>Default: unset. Overridden by the value from the map and the value in the batch properties file.</p>
<code>gen.definitions.WSDL</code>	<p>Boolean - Enable generation of WSDL definitions.</p> <p>Default: unset. Overridden by the value from the map and the value in the batch properties file.</p>
<code>gen.definitions.XMLPath</code>	<p>String - The base path for generated XML definitions.</p> <p>Default: unset. Overridden by the value from the map and the value in the batch properties file.</p>
<code>gen.definitions.DTDPPath</code>	<p>String - The base path for generated DTD definitions.</p> <p>Default: unset. Overridden by the value from the MapMaker config file, the value in the map and the value in the batch properties file.</p>
<code>gen.definitions.XMLSchemaPath</code>	<p>String - The base path for generated XML Schema definitions.</p> <p>Default: unset. Overridden by the value from the MapMaker config file, the value in the map and the value in the batch properties file.</p>



**Property Name****Description**`gen.definitions.WSDLPath`

String - The base path for generated WSDL definitions.

Default: unset. Overridden by the value from the MapMaker config file, the value in the map and the value in the batch properties file.

**Supplemental Directory Properties****Property Name****Description**`gen.supplemental.directories`

String - A colon or semi-colon separated list of additional directories to be included in the generated service JAR file.

Default: unset. Overridden by the value from the map and the value in the batch properties file.

**Deployment Properties****JI Service Deployment Properties****Property Name****Description**`deploy.jiSvc.jiJar`

Boolean - Deploy the JI service JAR file to the JI ResSvr if true.

Default: unset. Overridden by the value in the batch properties file.

`deploy.jiSvc.jiJar.jarPath`

String - The path to the JI service JAR file to be deployed.

Default: unset. Overridden by the value in the batch properties file.

Property Name	Description
<code>deploy.jiSvc.jiJar.mapPackage.name</code>	<p>String - The name of the map, including the package name.</p> <p>Default: unset, overridden by the value from the map and the value in the batch properties file.</p>
<code>deploy.jiSvc.jiJar.svc.names</code>	<p>String - A comma separated list of service names to be deployed.</p> <p>Default: unset. Overridden by the value from the map and the value in the batch properties file.</p>
<code>deploy.jiSvc.jiSvcBeanJar</code>	<p>Boolean - Deploy the customized EAServiceBean class for this service if true.</p> <p>Default: unset. Overridden by the value in the batch properties file.</p>
<code>deploy.jiSvc.JISvcBeanJar.jarPath</code>	<p>String - The path to the EAServiceBean to be deployed.</p> <p>Default: unset. Overridden by the value in the batch properties file.</p>
<code>deploy.jiSvc.JISvcBeanJar.svcPackage.name</code>	<p>String - The path to the extended EAServiceBean including the package name.</p> <p>Default: unset. Overridden by the value in the batch properties file.</p>
<code>deploy.jiSvc.resSvrUrl</code>	<p>String - The URL of the ResSvr the service JAR should be deployed to.</p> <p>Default: unset. Overridden by the value from the MapMaker config file and the value in the batch properties file.</p>

**Property Name****Description**`deploy.jiSvc.databaseUrl`

String - The URL of the database the service JAR should be deployed to.

Default: `localhost:30000/access`, overridden by the value in the batch properties file.

`deploy.jiSvc.aclAccount`

String - The account name to use when deploying the service JAR if ACL is enabled.

Default: unset. Overridden by the value in the batch properties file.

`deploy.jiSvc.aclPassword`

String - The account password to use when deploying the service JAR if ACL is enabled.

Default: unset. Overridden by the value in the batch properties file.

`deploy.jiSvc.notifyServers`

Boolean - Notify the JI EnvMgr(s) that the deployed service should be reloaded if true.

Default: unset. Overridden by the value in the batch properties file.

**Web Service Deployment Properties****Property Name****Description**`deploy.webSvc.webAppServer  
.location`

String - The host:port of the application server the web service should be deployed to or undeployed from.

Default: unset. Overridden by the value from the map and the value in the batch properties file.

Property Name	Description
<code>deploy.webSvc.webAppServer.userId</code>	<p>String - The admin User ID to use when connecting to the application server.</p> <p>Default: unset. Overridden by the value in the batch properties file.</p>
<code>deploy.webSvc.webAppServer.password</code>	<p>String - The admin User password to use when connecting to application server.</p> <p>Default: unset. Overridden by the value in the batch properties file.</p>
<code>deploy.webSvc.deploy.svc.names</code>	<p>String - A comma separated list of service names to be deployed to the application server.</p> <p>Default: unset. Overridden by the value in the batch properties file.</p>
<code>deploy.webSvc.deploy.svc.location</code>	<p>String - The URL address of the system where the SOAP gateway is running.</p> <p>Default: unset. Overridden by the value from the map and the value in the batch properties file.</p>
<code>deploy.webSvc.undeploy.svc.names</code>	<p>String - A comma separated list of service names to be undeployed from the application server.</p> <p>Default: unset. Overridden by the service names from the map.</p>
<code>deploy.webSvc.undeploy.svc.location</code>	<p>String - The URL of the application server the web service should be deployed to or undeployed from.</p> <p>Default: unset. Overridden by the value from the map and the value in the batch properties file.</p>

**Deprecated perform file properties**

Property Name	Description
<code>action.deploy</code>	<p>Boolean - Deploy a JI service. Superseded by the <code>action.deploy.jiSvc</code> property.</p> <p>Default: unset. Overridden by the value in the batch properties file.</p>
<code>notifyServers</code>	<p>Boolean - Notify servers after completing deployment. Superseded by the <code>deploy.jiSvc.notifyServers</code> property.</p> <p>Default: <code>false</code>.</p>
<code>jarFile</code>	<p>String - JI service jar file to deploy. Superseded by the <code>deploy.jiSvc.jiJar.jarPath</code> property.</p> <p>Default: unset. Overridden by the value in the batch properties file.</p>
<code>mapPackage</code>	<p>String - Superseded by the <code>deploy.jiSvc.jiJar.mapPackage.name</code> property.</p> <p>Default: unset. Overridden by the value from the map and the value in the batch properties file.</p>
<code>resourceServer</code>	<p>String - Superseded by the <code>deploy.jiSvc.resSvrUrl</code> property.</p> <p>Default: unset. Overridden by the value from the MapMaker config file and the value in the batch properties file.</p>
<code>database</code>	<p>String - Superseded by the <code>deploy.jiSvc.databaseUrl</code> property.</p> <p>Default: unset. Overridden by the value in the batch properties file.</p>

Property Name	Description
<code>serviceNames</code>	<p>String - Superceded by the <code>deploy.jiSvc.jiJar.svc.names</code> property.</p> <p>Default: unset. Overridden by the value from the map and the value in the batch properties file.</p>
<code>aclAccount</code>	<p>String - Superceded by the <code>deploy.jiSvc.aclAccount</code> property.</p> <p>Default: unset. Overridden by the value in the batch properties file.</p>
<code>aclPassword</code>	<p>String - Superceded by the <code>deploy.jiSvc.aclPassword</code> property.</p> <p>Default: unset. Overridden by the value in the batch properties file.</p>
<code>debug</code>	<p>Boolean - Superceded by the <code>action.debug.enable</code> property.</p> <p><b>Note:</b> Verbose logging may also be enabled via the <code>-debug</code> command line switch.</p> <p>Default: <code>false</code>.</p>

**Note:** Boolean `"true"` values may be specified as: `"y"`, `"yes"` or `"true"`. Case is ignored. Any properties that are not explicitly set will have their default value. A boolean property set to any other value will be considered `"false"`.

**Note:** The `"#"` character may be used to begin a comment. Anything between the `"#"` character and the end of the line is ignored. Blank lines are also ignored.

## Examples

### Example 1: Specify a configuration file when starting MapMaker

The command:

```
ea_mapmaker -f C:\JI\mmconfig
```

would start the MapMaker GUI with the properties specified in the C:\JI\mmconfig file instead of the default config file. The config file will be created if it does not exist.

### Example 2: Generate a JI service and clients, compile and deploy the service.

The command:

```
ea_mapmaker -batch C:\JI\Maps\JISvc.props C:\JI\Maps\JISvc.map
```

would load the batch properties from C:\JI\Maps\JISvc.props, then load the C:\JI\Maps\JISvc.map map and process according to the properties specified in the map and batch properties file listed below:

```
# Update the map after loading it.
```

```
action.updateMap: true
```

```
# Generate a JI service.
```

```
action.generate.service: true
```

```
# Generate service definitions. The definitions enabled in the map
```

```
# will be generated.
```

```
action.generate.definitions: true
```

```
# Generate JI clients. The clients enabled in the map will be generated.
```

```
action.generate.client: true
```

```
# Deploy the JI service jar to the default resource database
```

```
# (localhost:3000/access).
```

```
#
```

```
action.deploy.jiSvc: true
```

```
deploy.jiSvc.jiJar: true
```

## See Also

ea\_cfgmgr  
ea\_sysmon  
ea\_mapplayer

Sample batch properties and command files may be found in the `examples/mapmaker/batch` directory of the JI installation.



## ea\_mapplayer

### Syntax

```
ea_mapplayer [-m | -map <mapFile> [-a | -autoStart | --autostart
    [-hide | --hide | -shy | --shy]]] [-p | -port | --port <portNum>]
ea_mapplayer -? | -h | -help | --help
ea_mapplayer -v | -version
```

### Parameters

-a, -autoStart, --autoStart	When used in conjunction with the -map option, instructs MapPlayer to automatically start playing the map specified in <mapFile>.
-?, -h, -help, --help	Causes help information to be written to the mapplayer_stdout.txt file, located in <JI_install_dir>/logs.
-hide, --hide, -shy, --shy	When used in conjunction with the -autoStart option, instructs MapPlayer to hide its graphical user interface (GUI) once the map file has been opened and started.
-m, -map <mapFile>	Specifies the map file to open on startup.
-p, -port, --port <portNum>	Specifies the port on which MapPlayer will listen for connections.
-v, -version, --version	Causes version information about MapPlayer to be written to the mapplayer_stdout.txt file, located in <JI_install_dir>/logs.

**Note:** Command line options are case insensitive.

## Description

Launches the JI Integration MapPlayer, which can be used to play maps that have been generated in MapMaker. When used to play maps, MapPlayer which functions as a host simulator. For more information about the MapMaker, see the *JI Integration User's Guide*.

## See Also

ea\_mapmaker

## ea\_ping

### Syntax

```
ea_ping <command_param> [-B | --dbHome <DB_install_dir>]
    [-c | --eaEnv <dir>] [-h | --host <host>] [-I | --dbUser <DB_user_ID>]
    [-i | --jiInstallDir <JI_install_dir>] [-P | --dbPass <DB_password>]
    [-p | --port <port> ] [-u | --userClassPath <User_classpath>]
    [-x | --debug]
ea_ping -? | -help | --help
ea_ping version | --version
```

### Parameters

#### Ping Command Parameters

dbsvr, --dbSvr	Ping the JI Database daemon.
envmgr, --envMgr	Ping the JI Environment Manager.
ressvr, --resSv	Ping the JI Resource Server.
tomcat, --tomcat [<Servlet_opt>]	Ping the Tomcat servlet engine. Where <servletOpt> is "-a <appName>" or "--appName <appName>" and <appName> may be one of:  EAI        To ping the HTTP Gateway JIWSVC    To ping the SOAP Gateway

#### JI Configuration Option Parameters

-c, --eaEnv <JI_config_dir>	Location of the JI configuration files.
-h, --host <Host>	Host address of the JI server.
-i, --jiInstallDir <JI_install_dir>	JI Installation Directory.

<code>-p, --port &lt;Port&gt;</code>	TCP port number of the JI server to ping.
<code>-u, --userClasspath &lt;User_classpath&gt;</code>	Set the Java user classpath.

### Resource Database Option Parameters

<code>B, --dbHome &lt;DB_install_dir&gt;</code>	Database server installation Directory.
<code>-P, --dbPass &lt;DB_password&gt;</code>	Database password.
<code>-I, --dbUser &lt;DB_user_ID&gt;</code>	Database User ID.

### Miscellaneous Option Parameters

<code>-, --help, --help</code>	Print help information.
<code>-v, --version</code>	Show version information.
<code>-x, --debug</code>	Print debug information.

## Description

This program pings the specified server component to determine if the component is running.

### Locating the Server Host and Port

If the `-h` (`--host`) and `-p` (`--port`) command line options are not used, localhost will be assumed for the host, and the port will be determined in the following way:

- Environment Manager
  - The value of the `EA_ENVMGR` parameters in *environment.ccf*.
  - The value of the `EA_ENVMGR` environment variable.
- Resource Server
  - The value of the `config.rmiport` parameter in *ressvr.cfg*, the default Resource Server configuration file. The location of the *ressvr.cfg* file is assumed to be the `<JI_install_dir>/config` directory.

- Resource Database Server
  - The value of the DB\_TCP\_PORT parameter in *environment.ccf*.
  - The value of the DB\_TCP\_PORT environment variable.
  - The default Resource Database server listen port (30000)
- Tomcat Web Application Server
  - The value of the Connector port property in the Tomcat server config file: *server.xml* located in the : *<JI\_install\_dir>/tomcat/conf* directory.

Java command. Because *ea\_ping* executes underlying Java programs to shut down the JI Integration environment, it must be able to locate the correct Java Virtual Machine (JVM) executable (*java* in UNIX or *java.exe* in Windows). The location of the JVM is located by using either:

- The value of the *javacmd* parameter in *envmgr.cfg*, the default Environment Manager configuration file. The location of the *envmgr.cfg* file is assumed to be the *<JI\_install\_dir>/config* directory.

The value of *<JI\_install\_dir>* is located by using one of the following:

- The *-i <JI\_install\_dir>* command line flag, or:
- The *EA\_ENV* environment variable must identify the location of the *environment.ccf* file, which contains a definition of *EA\_HOME*.
- If the *envmgr.cfg* file is not found or the *javacmd* parameter is not set, then *java* will be used as a default command line. If this option is used, the JVM executable must be in the *PATH* environment variable.

## See Also

*ea\_status*

## ea\_ressvr

### Syntax

```
ea_ressvr [-f <cfgFile>] [-g] [-i <installDir>] [-p <svrPortNum>]
          [-l <logFile>] [-n] [-r] [-s]
ea_ressvr -h
ea_ressvr -v
```

### Parameters

<code>-f &lt;cfgFile&gt;</code>	Identifies the name of the configuration file for the Resource Server. If this parameter is not used, the default value, <code>envmgr.cfg</code> , is used. Optionally, the path can be identified in this parameter as well. If no path is identified, the default value, <code>&lt;JI_install_dir&gt;/config</code> is used.
<code>-g</code>	Causes the Resource Server graphical interface to be launched along with the server.
<code>-, -h, -help</code>	Causes help information to be written to the <code>ressvr_stdout.txt</code> file, located in <code>&lt;JI_install_dir&gt;/logs</code> .
<code>-i &lt;installDir&gt;</code>	Identifies the path to the installation of JI Integration.
<code>-p &lt;svrPortNum&gt;</code>	Identifies the port for Remote Method Invoke (RMI) activity for the Resource Server and the Environment Manager. If the Resource Server and Environment Manager are running on the same machine, the value of this port can match the value for the <code>rmiport</code> parameter in the Environment Manager configuration file or the <code>-d rmiport</code> command line option for the Environment Manager. However, if the same RMI port is used for the Environment Manager and the Resource Server and the Resource Server goes down, the Environment Manager will have to be restarted. This option will override the value of the <code>rmiport</code> parameter in the Resource Server's configuration file.

<code>-l &lt;logFile&gt;</code>	Identifies the name of the logfile for the Resource Server. Optionally, the path can be identified in this parameter as well. If no path is identified, the default value <code>&lt;JI_install_dir&gt;/logs</code> is used.
<code>-#</code>	Determines the level of debug logging. Allowable values for # are 0 through 9, with 0 representing minimal logging and 9 representing the most verbose logging.
<code>-r</code>	Causes RMI debugging information to be logged to the log file.
<code>-s</code>	Run the Resource Server as a Windows service.
<code>-v, -V, -version</code>	Causes version information about the Resource Server to be written to the <code>ressvr_stdout.txt</code> file, located in <code>&lt;JI_install_dir&gt;/logs</code> .

**Note:** Only the first character of the command line parameter is significant. For example, “-version” is equivalent to “-v”.

## Description

Launches the JI Integration Resource Server. The Resource Server is a process that runs in the JI Integration runtime server environment and acts as a centralized resource provider, facilitating access to JI Integration server-side data such as information stored in the JI Integration Resource Database.

A configuration file is used to identify configuration parameters for the Resource Server. The default name of the configuration file, `ressvr.cfg`, can be overridden using the `-f config_file` command line option when the Resource Server is launched. This option can also be used to indicate the path to the configuration file; if no path is identified, the default path `<JI_install_dir>/config` is used. All command line options override the equivalent settings in the Resource Server configuration file.

For more information about the Resource Server, see the *JI Integration User's Guide*.

## See Also

`ea_envmgr`

## ea\_shutdown

### Syntax

```
ea_shutdown [<Command_opt>] [-a | --aclAccount | --aclUser <ACL_Account>]
  [-B | --dbHome <DB_install_dir>] [-c | --eaEnv <JI_config_dir>]
  [-h | --dbHost <DB_host>] [-I | --dbUser <DB_user_ID>]
  [-i | --jiInstallDir <JI_install_dir>] [-P | --dbPass <DB_password>]
  [-p | --rmiPort <RMI_port>] [-Q | dbPort <DB_port>]
  [-s | --envMgrPort <EnvMgr_port>] [-V | --verbose]
  [-w | --aclPass <ACL_password>] [-x | --debug]
ea_shutdown -? | -help | --help
ea_shutdown -v | --version
```

### Parameters

#### Shutdown Command Option Parameters

-D, --dbSvr	Shut down the JI database daemon.
-e, --envMgr	Shut down the JI Environment Manager.
-j, --tomcat	Shut down the JI Jakarta Tomcat server.
-r, --resSvr	Shut down the JI Resource Server.

#### JI Configuration Option Parameters

-a <ACL_account>, --aclAccount <ACL_account>, --aclUser <ACL_account>	Specify the ACL account name.
-c <JI_config_dir>, --eaEnv <JI_config_dir>	The directory containing configuration files.
-i <JI_install_dir>, --jiInstallDir <JI_install_dir>	The JI Integration installation directory.
-p <RMI_port>, --rmiPort <RMI_port>	The Resource Server RMI port.



<code>-s &lt;EnvMgr_port&gt;, --envMgrPort &lt;EnvMgr_port&gt;</code>	The Environment Manager server port.
<code>-w &lt;ACL_password&gt;, --aclPass &lt;ACL_password&gt;</code>	Specify the ACL password.

### Resource Database Option Parameters

<code>-B &lt;Db_install_dir&gt;, --dbHome &lt;DB_install_dir&gt;</code>	Database server installation directory.
<code>-h &lt;DB_host&gt;, --dbHost &lt;DB_host&gt;</code>	Database server host address.
<code>-I &lt;DB_user_ID&gt;, --dbUser &lt;DB_user_ID&gt;</code>	Database User ID.
<code>-P &lt;DB_password&gt;, --dbPass &lt;DB_password&gt;</code>	Database Password.
<code>-Q &lt;DB_port&gt;, --dbPort &lt;DB_port&gt;</code>	Database server listen port.

### Miscellaneous Option Parameters

<code>-, -help, --help</code>	Print the help message.
<code>-V, --verbose</code>	Enable verbose logging.
<code>-v, --version</code>	Show version information.
<code>-x, --debug</code>	Enable debug logging.

### Description

This program is used to shut down components of the JI Integration server environment running on the local machine. Command line options that limit the components of the server environment that will be shut down (see "Shutdown Command Option Parameters" above) can be included when `ea_shutdown` is executed. These command line options can be also be combined to give complete control over the shutdown of server components. For example, if the Resource

Database server is shared among several environments and should not be shut down, the user may want to only shut down the Resource Server and Environment Manager. This can be done by executing the following command:

```
ea_shutdown -e -r
```

The `EA_ENV` environment variable must be set to the `<JI_install_dir>/config` directory or the `-c <JI_config_dir>` command line option must be used. The *environment.ccf*, *ressvr.cfg*, and *envmgr.cfg* files must be included in this directory. The *environment.ccf* file must contain a definition of `EA_HOME` as well as the `DB_TCP_PORT` parameter, which is required to shut down the Resource Database server. If the `-i <JI_install_dir>` command line flag is used, `EA_HOME` does not have to be set. Also, `EA_HOME` and `DB_TCP_PORT` can be defined as environment variables if the *environment.ccf* file is not used.

In order to execute the `ea_shutdown` command correctly, this program must be able to determine the port numbers of the server components and the location of the appropriate java command to use to execute the underlying java executables that shut down the various components of the environment. The program determines this information as follows:

- Port Numbers of the Server Components. The port numbers for each of the server components are determined as follows (in order of precedence):
  - Environment Manager  
Command line option (`-s <EnvMgr_port>`)  
The port entry in the Resource Database for the Environment Manager's configuration.  
Further, in order to ping the Environment Manager in order to determine that it is running, the value of the `EA_ENVMGR` parameter in the *environment.ccf* file should be changed to match the entry for the server port in the Resource Database. If this parameter is not set, `ea_shutdown` will look for the value of the `EA_ENVMGR` environment variable. If this is not found, it will use the default value for the Environment Manager port (30001).
  - Resource Server  
Command line option (`-p <RMI_port>`)  
The value of the `config.rmiport` parameter in *ressvr.cfg*, the default Resource Server configuration file. The location of the *ressvr.cfg* file is assumed to be the `<JI_install_dir>/config` directory.  
The default value for the Resource Server port (30002).
  - Resource Database Server  
Command line option (`-Q <DB_port>`)  
The value of the `DB_TCP_PORT` parameter in *environment.ccf*.  
The value of the `DB_TCP_PORT` environment variable.  
The default Resource Database server listen port (30000)
- Java command. Because `ea_shutdown` executes underlying Java programs to shut down the JI Integration environment, it must be able to locate the correct

Java Virtual Machine (JVM) executable (java in UNIX or java.exe in Windows). The location of the JVM is located by using either:

- The value of the `javacmd` parameter in `envmgr.cfg`, the default Environment Manager configuration file. The location of the `envmgr.cfg` file is assumed to be the `<JI_install_dir>/config` directory.
- If the `envmgr.cfg` file is not found or the `javacmd` parameter is not set, then java will be used as a default command line. If this option is used, the JVM executable must be in the PATH environment variable.

## See Also

`ea_start`

## ea\_shutsrvr

### Syntax

```
ea_shutsrvr <command> [-a | --aclAccount | --aclUser <aclUser>]
    [-c | --eaEnv <jiCfgDir>] [-i | --jiInstallDir <jiInstallDir>]
    [-p | --rmiPort <rmiPort>]
ea_shutdown --envMgr | envmgr
ea_shutdown --resSvr | ressvr
ea_shutsrvr -? | -help | --help
ea_shutsrvr -v | --version
```

### Parameters

#### Shutdown Command Parameters

dbsvr, --dbSvr	Shutdown the JI Resource Database server.
envmgr, --envMgr	Shutdown the JI Environment Manager.
ressvr, --resSvr	Shutdown the JI Resource Server.

#### JI Configuration Option Parameters

-a, --aclAccount, --aclUser <aclUser>	Specify the ACL account name or user ID.
-c, --eaEnv <jiCfgDir>	Specify the location of the JI configuration files.
-i, --jiInstallDir <jiInstallDir>	Specify the JI installation directory.
-p, --rmiPort <rmiPort>	Specify the RMI port for the JI Environment Manager and Resource Server.

<code>-s, --envMgrPort &lt;envMgrPort&gt;</code>	Specify the JI Environment Manager listen port.
<code>-w, --aclPass &lt;aclPass&gt;</code>	Specify the ACL password.

### Database Option Parameters

<code>-B, --dbHome</code>	Specify the JI Resource Database installation directory.
<code>-I, --dbUser &lt;dbUser&gt;</code>	Specify the JI Resource Database User ID.
<code>-P, --dbPass &lt;dbPass&gt;</code>	Specify the JI Resource Database password.
<code>-Q, --dbPort &lt;dbPort&gt;</code>	Specify the JI Resource Database listen port.

### Miscellaneous Option Parameters

<code>-, -help, --help</code>	Print the help message.
<code>-v, --version</code>	Print the program version and exit.
<code>-x, --debug</code>	Enable debug logging.

### Description

The `ea_shutsrvr` command is used by the `ea_shutdown` command to shut down the JI Integration Resource Database server, Resource Server and Environment Manager.

**Note:** The `ea_shutdown` command should be used for normal JI environment shutdowns.

### See Also

`ea_shutdown`

## ea\_start

### Syntax

```
ea_start [<Command_opt>] [-A | --aclAccount | --aclUser <ACL_account>]
    [-a | --dbBase <DB_data_dir>] [-B | --dbHome <DB_install_dir>]
    [-c | --eaEnv <JI_config_dir>] [-d | --daemon]
    [-I | --dbUser <DB_user_ID>] [-i | --jiInstallDir <JI_install_dir>]
    [-n | --retries <retries>] [-P | --dbPass <DB_password>]
    [-p | --rmiPort <RMI_port>] [-Q | --dbPort <DB port>]
    [-s | --envMgrPort <EnvMgr_port>] [-t | --sleepTime <seconds>]
    [-V | --verbose] [-w | --aclPass <ACL_password>] [-x | --debug]
ea_start -? | -help | --help
ea_start -v | --version
```

### Parameters

#### Start Command Option Parameters

-D, --dbSvr	Start the JI Resource Database server.
-e, --envMgr	Start the JI Environment Manager.
-j, --tomcat	Start the JI Jakarta Tomcat web application server.
-r, --resSvr	Start the JI Resource Server.

#### JI Configuration Option Parameters

-A <ACL_account>, --aclAccount <ACL_account>, --aclUser <ACL_account>	Specify the ACL account name.
-c <JI_config_dir>, --eaEnv <JI_config_dir>	The directory containing configuration files.
-i <JI_install_dir>, --jiInstallDir <JI_install_dir>	The JI Integration installation directory.

<code>-p &lt;RMI_port&gt;, --rmiPort &lt;RMI_port&gt;</code>	The Resource Server RMI port.
<code>-s &lt;EnvMgr_port&gt;, --envMgrPort &lt;EnvMgr_port&gt;</code>	The Environment Manager server port.
<code>-w &lt;ACL_password&gt;, --aclPass &lt;ACL_password&gt;</code>	Specify the ACL password.

### Resource Database Option Parameters

<code>-a &lt;Db_data_dir&gt;, --dbBase &lt;DB_data_dir&gt;</code>	Database server data directory.
<code>-B &lt;Db_install_dir&gt;, --dbHome &lt;DB_install_dir&gt;</code>	Database server installation directory.
<code>-I &lt;DB_user_ID&gt;, --dbUser &lt;DB_user_ID&gt;</code>	Database User ID.
<code>-P &lt;DB_password&gt;, --dbPass &lt;DB_password&gt;</code>	Database Password.
<code>-Q &lt;DB_port&gt;, --dbPort &lt;DB_port&gt;</code>	Database server listen port.

### JI Server Startup Option Parameters

<code>-n &lt;Ping_retries&gt;, --retries &lt;Ping_retries&gt;</code>	The number of times that ea_start will check to see if a server component has started. The default setting is 10.
<code>-t &lt;Seconds&gt;, --sleepTime &lt;Seconds&gt;</code>	The number of seconds that ea_start will wait before rechecking if a server component has started.

### Miscellaneous Option Parameters

<code>-, -help, --help</code>	Print help information.
-------------------------------	-------------------------

<code>-d, --daemon</code>	Start the server as a daemon.
<code>-V, --verbose</code>	Enable verbose logging.
<code>-v, --version</code>	Print version information.
<code>-x, --debug</code>	Enable debug logging.

## Description

This program is used to start components of the JI Integration server environment running on the local machine. Command line options that limit the components of the server environment that will be started (see "Start Command Option Parameters" above) can be included when `ea_start` is executed. These command line options can be also be combined to give complete control over startup of server components. For example, if the Resource Database server is shared among several environments and is already running, the user may need to only start the Resource Server and Environment Manager. This can be done by executing the following command:

```
ea_start -e -r
```

The `EA_ENV` environment variable must be set to the

`<JI_install_dir>/config` directory or the

`-c <JI_config_dir>` command line option must be used. The *environment.ccf*, *ressvr.cfg*, and *envmgr.cfg* files must be included in this directory. The *environment.ccf* file must contain a definition of `EA_HOME` as well as the `DB_TCP_PORT` parameter, which is required to start the Resource Database server. If the `-i <JI_install_dir>` command line flag is used, `EA_HOME` does not have to be set. Also, `EA_HOME` and `DB_TCP_PORT` can be defined as environment variables if the *environment.ccf* file is not used.

In order to execute the `ea_start` command correctly, this program must be able to determine the port numbers of the server components and the location of the appropriate java command to use to execute the underlying java executables that shut down the various components of the environment.

The program determines this information as follows:

- Port Numbers of the Server Components. The port numbers for each of the server components are determined as follows (in order of precedence):
  - Environment Manager  
Command line option (`-s <EnvMgr_port>`)  
The port entry in the Resource Database for the Environment Manager's configuration.  
Further, in order to ping the Environment Manager in order to determine that it is running, the value of the `EA_ENVMGR` parameter in the



*environment.ccf* file should be changed to match the entry for the server port in the Resource Database. If this parameter is not set, *ea\_start* will look for the value of the *EA\_ENVMGR* environment variable. If this is not found, it will use the default value for the Environment Manager port (30001).

- Resource Server  
Command line option (-p <RMI\_port>)  
The value of the *config.rmiport* parameter in *ressvr.cfg*, the default Resource Server configuration file. The location of the *ressvr.cfg* file is assumed to be the <JI\_install\_dir>/config directory.  
The default value for the Resource Server port (30002).
- Resource Database Server  
The value of the *DB\_TCP\_PORT* parameter in *environment.ccf*.  
The value of the *DB\_TCP\_PORT* environment variable.  
The default Resource Database server listen port (30000).
- Java command. Because *ea\_start* executes underlying Java programs to start the JI Integration environment, it must be able to locate the correct Java Virtual Machine (JVM) executable (java in UNIX or java.exe in Windows). The location of the JVM is located by using either:
  - The value of the *javacmd* parameter in *envmgr.cfg*, the default Environment Manager configuration file. The location of the *envmgr.cfg* file is assumed to be the <JI\_install\_dir>/config directory.
  - If the *envmgr.cfg* file is not found or the *javacmd* parameter is not set, then java will be used as a default command line. If this option is used, the JVM executable must be in the *PATH* environment variable.

## See Also

*ea\_shutdown*

## ea\_status

### Syntax

```
ea_status [-B | --dbHome <DB_install_dir>] [-c | --eaEnv <JI_config_dir>]
  [-e | --extended] [-I | --dbUser <DB_user_ID>]
  [-i | --jiInstallDir <JI_install_dir>] [-P | --dbPass <DB_password>]
  [-p | --rmiPort <RMI_port>] [-Q | --dbPort <DB_port>]
  [-s | --envMgrPort <EnvMgr_port>] [-x | --debug]
ea_status -? | -help | --help
ea_status -v | --version
```

### Parameters

#### JI Configuration Option Parameters

<code>-c &lt;JI_config_dir&gt;, --eaEnv &lt;JI_config_dir&gt;</code>	The directory containing configuration files.
<code>-i &lt;JI_install_dir&gt;, --jiInstallDir &lt;JI_install_dir&gt;</code>	The JI Integration installation directory.
<code>-p &lt;RMI_port&gt;, --rmiPort &lt;RMI_port&gt;</code>	The Resource Server's RMI port.
<code>-s &lt;EnvMgr_port&gt;, --envMgrPort &lt;EnvMgr_port&gt;</code>	The Environment Manager server port.

#### Resource Database Option Parameters

<code>-B &lt;Db_install_dir&gt;, --dbHome &lt;DB_install_dir&gt;</code>	Database server installation directory.
<code>-I &lt;DB_user_ID&gt;, --dbUser &lt;DB_user_ID&gt;</code>	Database User ID.
<code>-P &lt;DB_password&gt;, --dbPass &lt;DB_password&gt;</code>	Database Password.

-Q <DB\_port>,                      Database server listen port.  
--dbPort <DB\_port>

## Miscellaneous Option Parameters

-?, -help, --help              Print help information.

-e, --extended                Print extended status.

-v, --version                 Prints version information about *ea\_status*.

-x, --debug                  Print debug information.

## Description

This program pings the JI Tomcat web application server, Environment Manager, Resource Server and/or the Resource Database server and prints the results.

The EA\_ENV environment variable must be set to the

<JI\_install\_dir>/config directory or the

-c <JI\_config\_dir> command line option must be used. The *environment.ccf*, *ressvr.cfg*, and *envmgr.cfg* files must be included in this directory. The *environment.ccf* file must contain a definition of EA\_HOME as well as the DB\_TCP\_PORT parameter, which is required to ping the Resource Database server.

If the -i <JI\_install\_dir> command line flag is used, EA\_HOME does not have to be set. Also, EA\_HOME and DB\_TCP\_PORT can be defined as environment variables if the *environment.ccf* file is not used.

**Java command.** Because *ea\_status* executes underlying Java programs to shut down the JI Integration environment, it must be able to locate the correct Java Virtual Machine (JVM) executable (*java* in UNIX or *java.exe* in Windows). The location of the JVM is located by using either:

- The value of the javacmd parameter in *envmgr.cfg*, the default Environment Manager configuration file. The location of the *envmgr.cfg* file is assumed to be the <JI\_install\_dir>/config directory.
- If the *envmgr.cfg* file is not found or the javacmd parameter is not set, then java will be used as a default command line. If this option is used, the JVM executable must be in the PATH environment variable.
- **Port Numbers of the Server Components.** The port numbers for each of the server components are determined as follows (in order of precedence):
  - Environment Manager  
Command line option (-s <EnvMgr\_port>)

The port entry in the Resource Database for the Environment Manager's configuration.

Further, in order to ping the Environment Manager in order to determine that it is running, the value of the EA\_ENVMGR parameter in the *environment.ccf* file should be changed to match the entry for the server port in the Resource Database. If this parameter is not set, *ea\_status* will look for the value of the EA\_ENVMGR environment variable. If this is not found, it will use the default value for the Environment Manager port (30001).

- Resource Server

Command line option (-p <RMI\_port>).

The value of the *config.rmiport* parameter in *ressvr.cfg*, the default Resource Server configuration file. The location of the *ressvr.cfg* file is assumed to be the <JI\_install\_dir>/config directory.

The default value for the Resource Server port (30002).

- Resource Database server

The value of the DB\_TCP\_PORT parameter in *environment.ccf*.

The value of the DB\_TCP\_PORT environment variable.

The default Resource Database server listen port (30000).

## See Also

ea\_admin

ea\_ping

## ea\_sysmon

### Syntax

```
ea_sysmon [-f <cfgFile>] [-i <installDir>] [-n]
ea_sysmon -h
ea_sysmon -v
```

### Parameters

-f <cfgFile>	Identifies the name of the configuration file for the System Monitor. If this parameter is not used, the default value, <i>sysmon.cfg</i> , is used. Optionally, the path can be identified in this parameter as well. If no path is identified, the default value, <JI_install_dir>/config is used
-, -h, -H	Causes help information to be written to the <i>sysmon_stdout.txt</i> file, located in <JI_install_dir>/logs.
-i <installDir>	Identifies the path to the root directory of the installation of JI Integration.
-#	Determines the level of debug logging. Allowable values for # are 0 through 9, with 0 representing minimal logging and 9 representing the most verbose logging.
-v, -V	Causes version information about the System Monitor to be written to the <i>sysmon_stdout.txt</i> file, located in <JI_install_dir>/logs.

### Description

Launches the JI Integration System Monitor, used to monitor your JI Integration environment. A configuration file is used to identify configuration parameters for the System Monitor. The default name of the configuration file, *sysmon.cfg*, can be overridden using the -f config\_file command line option when the System Monitor is launched. This option can also be used to indicate the path to the configuration file; if no path is identified, the default path, <JI\_install\_dir>/config is used. All command line options override the equivalent settings in the System Monitor configuration file.

For more information about the System Monitor, see the *JI Integration User's Guide*.

## **See Also**

ea\_cfgmgr  
ea\_mapmaker

## ji\_svc

### Syntax

```
ji_svc -c | -i | -r | -v [-s <svcName>] [-d <displayName>]
      [-w <workingDirectory>] [-u <userName>] [-p <password>] [-l <logfile>]
ji_svc -? | -help
ji_svc -v
```

### Parameters

#### JI Windows Service Command Parameters

-c	Run <code>ji_svc</code> as a console application and start the JI Integration environment. This option is provided for testing purposes.
-?, -help	Print the help message.
-i	Install JI Integration as a Windows Service.
-r	Remove a JI Integration Windows Service installation.
-v	Print the <code>ji_svc</code> version and exit.

#### JI Windows Service Configuration Option Parameters

-d displayName	The name to be displayed in the services applet. The default display name is "JI Integration Service".
-l logfile	<p>The path of the logfile for the <code>ji_svc</code> command.</p> <p>If the <code>-l</code> option is not used to set the logfile path, but the environment variable <code>TMP</code> or <code>TEMP</code> is set, the default logfile path is <code>JISvc.log</code> in the directory set in <code>TMP</code> or <code>TEMP</code>.</p> <p>If the <code>-l</code> option is not used to set the logfile path and neither <code>TMP</code> nor <code>TEMP</code> is set, logging is disabled.</p>

<code>-p password</code>	The password used when stopping JI Integration with ACL enabled. The default password is "NTService".
<code>-s svcName</code>	The name used to identify the Windows Service. The default service name is "JIService".
<code>-u userName</code>	The user name used when stopping JI Integration with ACL enabled. The default user name is "NTService".
<code>-w workingDir</code>	The working directory for the JI Integration Windows Service.

## Description

The `ji_svc` command can be used to install or remove JI Integration as a Windows Service.

When run with the `-c` command line option, `ji_svc` starts the JI Integration environment. This option is provided to test the validity of other command line option arguments before installing JI Integration as a Windows Service. To stop JI Integration after starting it with the `ji_svc -c` command, run `ea_shutdown` from another command window and then interrupt the `ji_svc` command with `Control-C`.

When run with the `-i` command line option, `ji_svc` installs JI Integration as a Windows Service, allowing it to be controlled by the Windows Service Control Manager (SCM). The `-s` command line option allows you to change the name used when installing JI Integration as a Windows Service, while the `-d` option allows you to change the name displayed in the Windows **Services** applet.

When run with the `-r` command line option, `ji_svc` removes a previously installed Windows Service. If you have changed the default service name or have more than one instance of JI Integration installed as Windows Service, you need to specify the name of the service you wish to remove with the `-s` command line option.

When run with the `-v` command line option, `ji_svc` prints its version number and exits.

If after starting JI Integration as a Windows Service the environment is shut down with the `ea_shutdown` command, the **Service** applet still indicates that the JI Integration Windows Service is running. You can restart the environment with the `ea_start` command, or you can stop the JI Integration Windows service and restart it via the **Services** applet.



## Examples

The following table provides examples of everyday JI Integration operations performed via the `ji_svc` command line.

Operation	Command
Install JI Integration as a Windows Service, using C:\JI as the JI Integration installation directory and the default names.	<code>ji_svc -i -w C:\JI</code>
Remove the JI Integration Windows Service installed above.	<code>ji_svc -r</code>
Install two instances of JI Integration as Windows Services, with the names <code>JI45Dev</code> and <code>JI45Production</code> and corresponding display names.	<ul style="list-style-type: none"> <li><code>ji_svc -i -w C:\JI45DEV -s JI45Dev -d "JI 45 Development"</code></li> <li><code>ji_svc -i -w C:\JI45PROD -s JI45Prod -d "JI 45 Production"</code></li> </ul> <p><b>Note:</b> It is recommended to install each JI Integration Service instance in a separate directory.</p>
Remove the JI Integration Windows Services installed above.	<ul style="list-style-type: none"> <li><code>ji_svc -r -s JI45Dev</code></li> <li><code>ji_svc -r -s JI45Prod</code></li> </ul>
Run a console test start of JI Integration with ACL, using JIAdmin as the ACL account and JIPassword as the password.	<code>ji_svc -c -w C:\JI45 -u JIAdmin -p JIPassword</code>
Install the Windows Service tested above.	<code>ji_svc -i -w C:\JI45 -u JIAdmin -p JIPassword</code>
Install JI Integration as a Windows Service with logging enabled for the <code>ji_svc</code> command.	<code>ji_svc -i -w C:\JI45 -l C:\JI45\logs\ji_svc.log</code>



## Chapter 2. JI Integration GBBasic Reference

---

The Green Beans Basic (GBBasic) Java library is used to customize JI Integration Java services that have been generated in MapMaker. For more information about using custom classes with JI Integration services, see the MapMaker documentation in the *JI Integration User's Guide*.

Information on classes and interfaces is provided in HTML format in JavaDoc form. JavaDoc for JI Integration Custom Classes is included with your JI Integration installation. The JavaDoc, which can be opened in any Web browser, is located in `<JI_install_dir>/html/javadoc.html`.

View GBBasic API Reference.

The following packages are included with the GBBasic library:

- Package `com.jacada.mapstudio.gbbasic`
- Package `com.jacada.mapstudio.gbbasic.event`



## Chapter 3. JI Integration EAServiceBean Reference

---

The EAServiceBean package (com.jacada.ea.ejb.server) represents the service interface between JI Integration service code and JServices in the JI Integration server environment.

Information on classes and interfaces is provided in HTML format in JavaDoc form. JavaDoc for JI Integration Custom Classes is included with your JI Integration installation. The JavaDoc, which can be opened in any Web browser, is located in `<JI_install_dir>/html/javadoc.html`.

View EAServiceBean API Reference

The EAServiceBean package can be customized if required. For information about customizing JI Integration service code (rather than the EAServiceBean interface), see Chapter 2, "JI Integration GBBasic Reference".

The following package is included with the JI Integration EAServiceBean:

- Package com.jacada.ea.ejb.server



## Chapter 4. JClient3 API Reference

---

Information on classes and interfaces included in the Java Client Library version 3 is provided in HTML format in JavaDoc form. JavaDoc for JI Integration Custom Classes is included with your JI Integration installation. The JavaDoc, which can be opened in any Web browser, is located in `<JI_install_dir>/html/javadoc.html`.

View JClient3 API Reference.

The following packages are included with the JClient3 library:

- Package `com.jacada.ea.jclient3`
- Package `com.jacada.ea.jclient3.utility`





## Chapter 5. JI Integration Error Messages

---

The following table lists the possible JI Integration messages, describes the probable cause, and identifies the action to correct the problem.

<b>EA_ errno</b>	<b>EAErrStr</b>	<b>#define</b>	<b>Description</b>
0	No error	EA_OK	Normal return status.
101	General Error from Environment Manager	EA_MGR_ERROR	Error returned from the Environment Manager for a number of problems.
102	Message version mismatch - recompile	EA_COMM_VERSION_ERROR	Message sent to the current process does not match the message format required. This can happen when older versions of JI Integration clients try to connect to the Environment Manager.
103	Application name already registered	EA_REGISTER_ERROR	The referenced application has already been registered.
104	General error	EA_GENERAL_ERROR	Error returned for a number of infrastructure problems.
105	Syntax error in FLDDDESC	EA_SYNTAX_ERROR	Field description argument is an invalid format.
106	Could not find field using FLDDDESC	EA_FIELD_ERROR	Specified field was not found.

<b>EA_ errno</b>	<b>EAErrStr</b>	<b>#define</b>	<b>Description</b>
107	New type does not match old type	EA_TYPE_ERROR	Specified type does not match a previous type.
108	Illegal type field	EA_ILLEGAL_TYPE_ERROR	Requested field has an unknown type. This is an internal error.
109	Illegal request was made of JI Integration	EA_ILLEGAL_REQUEST	Invalid request was made. For example, using EA_CIntSvcNBInvoke() on a service that did not respond to a previous invoke.
110	Illegal parameter value was given	EA_PARAMETER_ERROR	Invalid parameter was passed to a JI Integration function.
111	Type of TRXDS does not match expected	EA_TRXDS_TYPE_ERROR	Invalid screen or service message argument. This is an internal error.
112	Communication error	EA_COMM_ERROR	Error occurred while reading, writing, or setting up a socket.
113	Message queue error	EA_MSGQ_ERROR	Error occurred while accessing a message queue. Message queues are used to communicate to Protocol Agents and the Monitor Agent.
114	Unexpected NULL field encountered	EA_NULL_FLD	Null field was encountered where data was expected. This is an internal error.
115	EA_CIntLogin has not been called	EA_IS_NOT_OPEN	The client login was executed with error. The client should log in before any other client API.

<b>EA_ errno</b>	<b>EAErrStr</b>	<b>#define</b>	<b>Description</b>
116	JI Integration is already logged in	EA_IS_OPEN	The client has logged in already and is trying to log in again.
117	Bad key must be W R B V or I	EA_BAD_KEY	Bad field key found. This is an internal error.
118	Behavior prevents operation because of key	EA_KEY_LOCK	Key field value prevents operation on field.
119	File error	EA_FILE_ERROR	General file access error.
120	Cannot get memory to add information	EA_MEMORY_ALLOC_ERROR	<i>malloc</i> failed.
121	Area reference to non-area	EA_AREA_REF_ERROR	Specified field is treated as an area in screen or service message but is not.
122	Field name too long	EA_LENGTH_ERROR	Length of specified field name too long.
123	Field does not exist	EA_FIELD_EXIST_ERROR	Specified field does not exist in the screen or service message.
124	End of fields found	EA_END_OF_FIELDS	No more fields with this name exist in the screen or service message.
125	Client handle not found	EA_NO_SUCH_APPL	Specified client handle is not valid. This is an internal error.
126	Field or tag off the screen	EA_OFF_SCREEN	Defined screen tag is mapped outside the current screen boundaries.

<b>EA_ errno</b>	<b>EAErrStr</b>	<b>#define</b>	<b>Description</b>
127	Illegal command	EA_ILLEGAL_COMMAND_ERROR	Command contained in the message is not valid. This is an internal error.
128	Bad header parameters	EA_HEADER_ERROR	Values in the current message are not valid. This is an internal error.
129	Service does not exist	EA_NO_SUCH_SVC	Specified service handle does not exist or is invalid.
130	Environment Manager cannot perform request	EA_NO_FACILITIES_AVAIL	Environment Manager cannot perform the request because of a defined limit. The maximum number of services, clients, or Protocol Agents was reached.
131	Time-out trying to start PA	EA_PA_TIMEOUT	Environment Manager timed out waiting for the Protocol Agent to make its connection.
132	No appropriate PA available	EA_NO_AVAIL_PAS	Environment Manager could not start the specified Protocol Agent.
133	PA does not exist	EA_NO_SUCH_PA	Specified handle does not match any known Protocol Agent.
134	PA type not ANY SYNC ASYNC TN3270	EA_BAD_PA_TYPE	Specified Protocol Agent is not one of the acceptable types.
135	Invalid PA handle	EA_BAD_PA_PTR	Internal structure is not valid for the Protocol Agent. This is an internal error.

<b>EA_ errno</b>	<b>EAErrStr</b>	<b>#define</b>	<b>Description</b>
136	Service could not start properly	EA_SVC_STARTUP_ERROR	Number of causes, including system resource limits or Resource Server problems.
137	Service could not attach PA	EA_SVC_PA_ATTACH_ERROR	Service is having a problem establishing communication with a Protocol Agent.
138	PA handle is attached or does not exist	EA_PA_NOT_AVAILABLE	Service assigned a Protocol Agent already in use. This is an internal error.
139	Current Service status is BUSY	EA_SVC_BUSY	The service is currently busy.
140	Current Service status is IDLE	EA_SVC_IDLE	The service is currently idle.
141	Could not open sequence file	EA_BAD_SEQ_FILENAME	The sequence file could not be opened.
142	Could not match required context	EA_BAD_CONTEXT	Current presentation space does not match any of the specified screens.
143	Function not supported	EA_NOT_SUPPORTED	Request made is not supported by the type of Protocol Agent.
144	Specified screen/field entry is not in the Resource Database	EA_NOT_IN_DB	Specified screen could not be found in the Resource Database.
145	General time-out error	EA_TIMEOUT_ERROR	Timeout occurred while waiting for a response.

<b>EA_errno</b>	<b>EAErrStr</b>	<b>#define</b>	<b>Description</b>
147	Tags do not match in EA_ScreenGet.	EA_BAD_TAG_MATCH	Screen read is performed with a screen mask that does not match the current presentation space.
148	Host appears down	EA_HOST_DOWN	Two calls to EA_ScreenWait() are performed back-to-back with no data sent from the host the second time. This error is generated by the Protocol Agent that did not receive the data from the host.
149	Settling time expired	EA_SETTLE_TIMEOUT	Screen settle time was reached before the cursor landed in the settle position.
150	Exceeded max services for this client	EA_MAX_SVCS	Maximum number of services are currently open by this client.
151	Exceeded maximum PAs for this client	EA_MAX_PAS	Maximum number (20) of Protocol Agents are currently open by this service.
152	Error building a TRXDS	EA_TRXDS_BUILD_ERROR	Error building an internal TRXDS message. This is an internal error.
153	General Resource Database error	EA_DBLIB_ERROR	Problem occurred while using the Resource Database.
154	General shared memory error	EA_SHM_ERROR	Problem occurred using shared memory.

<b>EA_ errno</b>	<b>EAErrStr</b>	<b>#define</b>	<b>Description</b>
155	PA can not open terminfo file	EA_NON_OPEN_TI_FILE	Protocol agent could not open the <i>terminfo</i> file in <i>&lt;EA_HOME/lib/ProtoAgents/config</i> for the Telnet Protocol Agent.
156	PA unsupported terminal type	EA_UNSUPPORTED_TERM	Specified terminal type is not supported for the Telnet Protocol Agent.
157	PA terminal type not found in terminfo file	EA_TERM_NOT_FOUND	Specified terminal type was not found in the <i>&lt;EA_HOME&gt;/lib/ProtoAgents/config/*.ti</i> file for the Telnet Protocol Agent.
158	PA host or terminal connect failure	EA_NON_OPEN_DEVICE	Problems occurred while connecting to the host or terminal device for the Telnet Protocol Agent.
159	PA bad read status from device	EA_TTY_READ_ERR	Error occurred while reading from the host or terminal device for the ASYNC or Telnet Protocol Agent.
160	PA bad write status from device	EA_TTY_WRITE_ERR	Error occurred while writing from either the host or terminal device for the ASYNC or Telnet Protocol Agent.
161	Error with ioctl in PA	EA_IOCTL_ERROR	An <i>ioctl</i> system call failed for the ASYNC or Telnet Protocol Agent.
162	Invalid parameter received from service	EA_INVALID_PARAM	Illegal argument was sent to an API call.

<b>EA_ errno</b>	<b>EAErrStr</b>	<b>#define</b>	<b>Description</b>
163	Maximum characters reached	EA_MAX_CHAR	Maximum number of characters was reached before one of the settle strings was hit. This error is generated by EA_ScreenScrollGet().
164	AsyncPA received unknown priority msg	EA_UNKNOWN_PM	Invalid priority message was sent to the Telnet Protocol Agent.
165	AsyncPA could not send data to device	EA_UNDELIVERABLE_KEYS	Key codes sent with EA_ScreenSendKeys() are invalid and cannot be sent to the host.
166	AsyncPA position cursor failure	EA_UNSUCCESSFUL_MOVE	Protocol agent failed to move the cursor to one of the writable fields. The Telnet Protocol Agent must be able to instruct the host to move the cursor to each field specified in the screen masks before they can write to the field. This error occurs if you cannot tab to the start of each writable field in the screen.
167	AsyncPA could not open/read config file	EA_NON_OPEN_CFG_FILE	Protocol agent could not open the proper configuration file in <EA_HOME>/lib/ProtoAgents/config. The files specified in the Resource Editor must be in this directory.



<b>EA_ errno</b>	<b>EAErrStr</b>	<b>#define</b>	<b>Description</b>
168	Environment Manager is shutting down	EA_IS_SHUTTING_DOWN	The Environment Manager is shutting down.
169	Proto Agent Died	EA_PA_DYING	The Protocol Agent is down.
170	Not a service	EA_NOT_A_SERVICE	API called is only valid for a JI Integration service. The call was made as a client, or the linking was done improperly.
171	Field already exists	EA_FIELD_ALREADY_EXIST	Specified field has already been created.
172	Service died	EA_SVC_DIED	The desired service has expired.
173	Service did not reply with a svcmsg	EA_SVC_NO_REPLY	A service message was not returned by the service named in the API call.
176	Internal hash table error	EA_HASHTABLE_ERROR	An error has occurred in the JI Integration internal databases. This is an internal error.
178	Break message was received	EA_BREAK_ERROR	A blocking service call, such as EA_ScreenMonitorKeys() was interrupted as a result of an EA_SvcBreak() call.
179	Client killed	EA_CLNT_KILLED	Client was killed.
180	Exceeded max PATERMS	EA_MAX_PATERMS	A PATERM was not created because the maximum number of PATERMs has already been created by the client/service

<b>EA_ errno</b>	<b>EAErrStr</b>	<b>#define</b>	<b>Description</b>
181	Protocol Agent Manager error	EA_PAMGR_ERROR	An error occurred in the Protocol Agent Manager. This is an internal error.
182	Writable PTerm not available	EA_PATERM_ERROR	A Read/WritePTerm was not available.
183	WIDESTRING mode not set	EA_WIDESTRING_ERROR	WIDESTRING mode was not set.
1001	API: Device/host not connected	EA_PA_NO_CONNECT	Connection to the host was lost for the TN3270 or TN5250 Protocol Agent.
1004	API: Presentation space busy	EA_PA_PS_BUSY_ERR	Function performed on a presentation space that is in transition. This error is returned from the HLLAPI library when the presentation space is locked and a send function was attempted for the TN3270 or TN5250 Protocol Agent.
1005	API: Presentation space locked	EA_PA_PS_LOCK_ERR	Timeout occurs while waiting for the host to be unlocked for the TN3270 or TN5250 Protocol Agent, for example, you are trying to write to a read-only field. This error locks the keyboard until a reset key is sent.
1006	API: User data truncated	EA_PA_TRUNCATE_ERR	Data written to the presentation space was lost for the TN3270 or TN5250 Protocol Agent. This error can occur is a mask field is defined larger than the real field in the presentation space.

<b>EA_errno</b>	<b>EAErrStr</b>	<b>#define</b>	<b>Description</b>
1007	API: Invalid cursor position	EA_PA_PS_POSITION_ERR	Cursor position command failed for the TN3270 or TN5250 Protocol Agent. This is an internal error.
1008	API: Start aid key intercept not executed	EA_PA_AIDKEY_INT_ERR	Start call was not called for the TN3270 or TN5250 Protocol Agent. EA_ScreenMonitorKeys() uses HLLAPI aid key intercept function. This is an internal error.
1009	API: Error detected by API	EA_PA_API_ERR	HLLAPI call failed for the TN3270 or TN5250 Protocol Agent. This is an internal error.
1010	API: TN3270 hip scheduler error	EA_PA_API_SCHED_ERR	Internal error for the TN3270 or TN5250 Protocol Agent.
1011	API: TN3270 Telnet connect error	EA_PA_API_TN_ERR	The TN3270 or TN5250 Protocol Agent fails to connect to a TN3270 server.
1024	API: String Search not Found	EA_PA_STR_NOT_FOUND_ERR	Search string was not found.
1025	API: Data Has not Changed in Last Minute	EA_PA_IPAUSE_ERR	An error occurred with the Protocol Agent because no data has changed in the last minute.

<b>EA_ errno</b>	<b>EAErrStr</b>	<b>#define</b>	<b>Description</b>
1027	API: Message queue overrun error	EA_PA_Q_OVER	Protocol agent cannot respond to the service because the normal return message does not fit in the message queue. This is for the TN3270 and TN5250 Protocol Agent.
2000	Invalid message command	NO_SUCH_COMMAND	Invalid command contained in the message.
2002	Couldn't start instance of service	COULD_NOT_INSTANTIATE	Could not instantiate the service
2003	Service class is not of acceptable type	INVALID_SERVICE_CLASS_TYPE	The service class is not valid.
2004	Service is not of a known type	INVALID_SERVICE_TYPE	Service type is not recognized.
2005	Service is disabled	SERVICE_DISABLED	The service has been disabled in the Configuration Manager.
2010	No service arguments given	NO_SERVICE_ARGUMENTS	Required arguments for the service were not passed from the client.
2011	No service master record configured	NO_SERVICE_MASTER_CONFIG	There is no service master record in the Resource Database for the requested service.
2012	No service detail record configured	NO_SERVICE_DETAIL_CONFIG	There is no service detail record in the Resource Database for the requested service. Applicable only if the Environment Manager configuration is set to accept connections for its configured service details only.

<b>EA_ errn o</b>	<b>EAErrStr</b>	<b>#define</b>	<b>Description</b>
2013	Can't load a class needed by the service	SERVICE_CANNOT_FIND_CLASS	A class file required by the service was not located and could not be loaded.
2014	No service context instance	NO_SERVICE_CONTEXT	No service context instance. This is an internal error.
2015	Service context is of incorrect type	INVALID_SERVICE_CONTEXT	Service context is of incorrect type. This is an internal error.
2030	Could not start service	ERROR_STARTING_SERVICE	General error indicating that the service could not be launched.
2040	Max service instances has been reached	MAX_INSTANCES_REACHED	The maximum service instance, defined in the Configuration Manager for either the service master or service detail, has been reached.
2041	Session name is already being used	SESSION_NAME_IN_USE	The service instance that uses the requested session name is already being used by a client, and cannot be used by another client until the current client disconnects.
2050	Could not connect to host	HOST_CONNECT_ERROR	There was an error connected to the legacy host. The host name or IP address and port of the legacy host is defined in MapMaker during trail recording or is set in the Configuration Manager.

<b>EA_ errno</b>	<b>EAErrStr</b>	<b>#define</b>	<b>Description</b>
2051	Could not disconnect from host	HOST_DISCONNECT_ERROR	There was an error disconnected to the legacy host.
2060	Could not invoke service	SERVICE_INVOCATION_ERROR	The Environment Manager could not invoke the requested service.
2200		INVALID_METHOD_ARGUMENT_COUNT	Invalid number of arguments for the method.
2201		INVALID_METHOD_ARGUMENT_TYPE	Invalid type of argument for the method.
2202		METHOD_INVOCATION_ERROR	The service failed to invoke the method.
2203		CANNOT_SET_METHOD_INPUT_VAR	The service was unable to set an input variable for the method.
2204		CANNOT_GET_METHOD_LIST	The service was unable to obtain a list of methods in the service.
2205		NO_SUCH_METHOD	The requested method does not exist in the current service.
2206		CANNOT_GET_METHOD_PARAMS	The service was unable to obtain a list of parameters for the method.
2300		EJB_CREATION_ERROR	An error occurred when the EAServiceBean was being created.

<b>EA_ errn o</b>	<b>EAErrStr</b>	<b>#define</b>	<b>Description</b>
2400		NO_SERVICE_ LICENSES_AVAIL	Either the environment is not correctly licensed for services, or the number of services exceeds the number allowed in the JI Integration license.





## Chapter 6. Keyboard Mapping

---

Because of the flexible nature of keyboard mapping for various legacy applications, it may be necessary to change the way that MapMaker interprets certain keystrokes that are input by the user. For example, because modern PC keyboards do not have PA keys, unique keystrokes are required to send the PA1, PA2, and PA3 keys. By default, these keys are mapped to the keystrokes `Control+t`, `Control+y`, and `Control+u` respectively. So to send the PA1 key to the mainframe application while using MapMaker, you would press `Control+t`.

Keyboard mapping for MapMaker is defined in ASCII text files. The default location of these files is `<JI_install_dir>/config` where `<JI_install_dir>` is the directory in which JI Integration was installed. The following default keyboard mapping files are included with JI Integration:

- *tn3270.kbm*
- *tn5250.kbm*

These files can be modified by the user, following directions in “Key Map File Structure” on page 110, below.

### Setting the Keyboard Map File in MapMaker

---

The keyboard map file can be set in MapMaker using the Properties dialog box. In MapMaker, select **File > Properties**. In the General tab, enter the path and filename for the keyboard mapping file for 3270 and 5250.

**Note:** The keyboard mapping files only need to be used if the default settings are changed. The default settings in the keyboard mapping files are used by default in MapMaker.

## TN3270 Keyboard Mapping

Emulation Function	Keystroke Sequence for 3270 Emulation
PF1	F1
PF2	F2
PF3	F3
PF4	F4
PF5	F5
PF6	F6
PF7	F7
PF8	F8
PF9	F9
PF10	F10
PF11	F11
PF12	F12
PF13	Shift+F1
PF14	Shift+F2
PF15	Shift+F3
PF16	Shift+F4
PF17	Shift+F5
PF18	Shift+F6

<b>Emulation Function</b>	<b>Keystroke Sequence for 3270 Emulation</b>
PF19	Shift+F7
PF20	Shift+F8
PF21	Shift+F9
PF22	Shift+F10
PF23	Shift+F11
PF24	Shift+F12
PA1	Control+t
PA2	Control+y
PA3	Control+u
Enter	Enter
Up	Up
Down	Down
Left	Left
Right	Right
Clear	Control+z
Home	Home
Reset	Control+r
Tab	Tab
Backtab	Shift+Tab

<b>Emulation Function</b>	<b>Keystroke Sequence for 3270 Emulation</b>
EEOF	Control+p
Del	Del
Attn	Control+a
SysReq	Control+s
Backspace	Backspace
Ins	Ins
Ins	Control+o
Cent	Control+c

## TN5250 Keyboard Mapping

Emulation Function	Keystroke Sequence for 5250 Emulation
F1	F1
F2	F2
F3	F3
F4	F4
F5	F5
F6	F6
F7	F7
F8	F8
F9	F9
F10	F10
F11	F11
F12	F12
F13	Shift+F1
F14	Shift+F2
F15	Shift+F3
F16	Shift+F4
F17	Shift+F5
F18	Shift+F6

<b>Emulation Function</b>	<b>Keystroke Sequence for 5250 Emulation</b>
F19	Shift+F7
F20	Shift+F8
F21	Shift+F9
F22	Shift+F10
F23	Shift+F11
F24	Shift+F12
Enter	Enter
Up	Up
Down	Down
Left	Left
Right	Right
RollUp	PgDn
RollDown	PgUp
Clear	Control+z
Home	Home
Reset	Control+r
Tab	Tab
Backtab	Shift+Tab
EEOF	Control+p

<b>Emulation Function</b>	<b>Keystroke Sequence for 5250 Emulation</b>
FldExit	Control+q
Help	Control+j
Del	Del
Backspace	Backspace
Ins	Ins
Ins	Control+o
Attn	Control+a
SysReq	Control+s
Cent	Control+c

## Key Map File Structure

The contents of the keyboard mapping files consist of name=value pairs that assign a 3270 or 5250 function to a keyboard key sequence as follows:

```
<emulation function>=<keyboard key sequence>
```

All text following a semi-colon (;) is treated as a comment.

```
; This is a comment
```

The following sections outline the allowable values for both the emulation name and the key sequences that can be configured in the keyboard mapping files.

### Emulation Function Names

The following three sections define the emulation functions that can be configured in the keyboard mapping files, and list the corresponding name that should be used as the emulation function name in the name=value pair configured in the keyboard mapping file.

#### TN3270 Emulation

The following 3270 emulation functions may be mapped:

Emulation Function	Name
Attention	Attn
Backspace	Backspace
Back Tab	Backtab
Clear	Clear
Cursor Left	Left
Cursor Right	Right
Cursor Up	Up
Cursor Down	Down
Delete	Del



<b>Emulation Function</b>	<b>Name</b>
Enter	Enter
Erase EOF	EEOF
Forward Tab	Tab
Home	Home
Insert	Ins
PA1 through PA3	PA1 through PA3
PF1 through PF24	PF1 through PF24
Reset	Reset
SYSREQ	SysReq

### **TN5250 Emulation**

The following 5250 emulation functions may be mapped:

<b>Emulation Function</b>	<b>Name</b>
Attention	Attn
Backspace	Backspace
Back Tab	Backtab
Clear	Clear
Cursor Left	Left
Cursor Right	Right
Cursor Up	Up

<b>Emulation Function</b>	<b>Name</b>
Cursor Down	Down
Delete	Del
Enter	Enter
Erase EOF	EEOF
Forward Tab	Tab
Help	Help
Home	Home
Insert	Ins
NewLn	NewLn
PA1 through PA3	PA1 through PA3
PF1 through PF24	PF1 through PF24
Reset	Reset
Roll Down	RollDn
Roll Up	RollUp
SYSREQ	SysReq

## Keyboard Sequences

The keyboard key sequences that may be assigned to an emulation function include control and function keys. The following table defines these keys:

Keyboard Function	Definition
Control keys	Control+ followed by 1 or more characters
Function keys	F1 through F12
Shift	Shift+ followed by 1 or more characters

### Control Keys

Control keys are defined by a ^ followed by a character sequence. The allowed values are:

**Valid control keys** • 'a' through 'z'

- f1 through f12
- home
- end
- page\_up
- page\_down
- del
- enter
- backspace
- '[', ']', '^', '\_'
- up
- down
- left
- right

## Examples

### Control Keys

- `Attn=Control+a`  
Assigns `Control-a` as the 3270 Attention key.
- `Ins=Control+Enter`  
Assigns `Control-Enter` as the 3270 Insert key.

### Other Keys

The following are some examples of key sequences other than control keys:

- `PF1=F1`  
Assign `F1` as the 3270 PF1 key.
- `Home=Home`  
Assign `Home` as the 3270 Home key.
- `EEOF=Ins`  
Assign `Ins` as the 3270 Erase EOF key
- `pf13+=F1`  
Assign `Shift-F1` as the 3270 PF13 key

## Protocol Agent Terminal Keyboard Mapping

---

Protocol Agent terminals are used to interface with legacy applications while debugging or monitoring JI Integration services. Because of the flexible nature of keyboard mapping for various legacy applications, it may be necessary to change the way that write-enabled Protocol Agent terminals interpret certain keystrokes that are input by the user. For example, because modern PC keyboards do not have PA keys, unique keystrokes are required to send the PA1, PA2, and PA3 keys. By default, these keys are mapped to the keystrokes `Control+t`, `Control+y`, and `Control+u` respectively. So to send the PA1 key to the mainframe application while using `PATerm`, you would press `Control+t`.

## Keyboard Definition

The Protocol Agent terminal keyboard mapping is defined in an ASCII text file called *paterm.kbm*, located in `<JI_install_dir>/lib/ProtoAgents` where `<JI_install_dir>` represents the directory into which JI Integration was installed. This file can be modified by the user, following directions in “Key Map File Structure” on page 118, below. The default layout of this file is as follows:

Function	Keys for TN3270	Keys for TN5250
Attention	Ctrl-A	Ctrl-A
Backtab	Ctrl-B, Shift-TAB	Ctrl-B, Shift-TAB
Cent Sign (¢)	Esc-c	Esc-c
Clear	Ctrl-Z	Ctrl-Z
Cursor Select	Ctrl-G	Ctrl-G
Cursor Left	..	..
Cursor Right	Æ	Æ
Cursor Up		
Cursor Down	Ø	Ø
Cursor Home	Ctrl-Q, Home	Ctrl-Q, Home
Delete	Ctrl-E, Del	Ctrl-E, Del
DUP	Ctrl-D	Ctrl-D
Enter	Enter	Enter
Erase EOF	Ctrl-P	Ctrl-P
Erase Input	Ctrl-X	Ctrl-X
Field +	--	Ctrl-L
Field -	--	Ctrl-K
Field Mark	Ctrl-J	--
Forward Tab	Tab, Ctrl-F	Tab, Ctrl-F

Function	Keys for TN3270	Keys for TN5250
Help	--	Ctrl-J
Hex	--	\ex
Home	Ctrl-Q, Home	Ctrl-V, Home
Insert	Ctrl-O, Ins	Ctrl-O, Ins
Logical Not (")	^	^
New Line	Ctrl-N	Ctrl-N
PA1	Ctrl-T	Ctrl-T
PA2	Ctrl-Y	Ctrl-Y
PA3	Ctrl-U	Ctrl-U
Reset	Ctrl-R	Ctrl-R
Roll Down	--	Page Up, Esc-Esc-u
Roll Up	--	Page Down, Esc-Esc-d
SYSREQ	Ctrl-S	Ctrl-S
Vertical Solid Bar	Ctrl-\	Ctrl-\
PF1/F1	F1, Esc-1	F1, Esc-1
PF2/F2	F2, Esc-2	F2, Esc-2
PF3/F3	F3, Esc-3	F3, Esc-4
PF4/F4	F4, Esc-4	F4, Esc-4
PF5/F5	F5, Esc-5	F5, Esc-5
PF6/F6	F6, Esc-6	F6, Esc-6
PF7/F7	F7, Esc-7, Page Up	F7, Esc-7
PF8/F8	F8, Esc-8, Page Down	F8, Esc-8
PF9/F9	F9, Esc-9, Ctrl-F1	F9, Esc-9, Ctrl-F1
PF10/F10	F10, Esc-0, Ctrl-F2	F10, Esc-0, Ctrl-F2

<b>Function</b>	<b>Keys for TN3270</b>	<b>Keys for TN5250</b>
PF11/F11	F11, Esc-q, Ctrl-F3	F11, Esc-q, Ctrl-F3
PF12/F12	F12, Esc-w, Ctrl-F4	F12, Esc-w, Ctrl-F4
PF13/F13	Shift-F1, Esc-e	Shift-F1, Esc-e
PF14/F14	Shift-F2, Esc-r	Shift-F2, Esc-r
PF15/F15	Shift-F3, Esc-t	Shift-F3, Esc-t
PF16/F16	Shift-F4, Esc-y	Shift-F4, Esc-y
PF17/F17	Shift-F5, Esc-u	Shift-F5, Esc-u
PF18/F18	Shift-F6, Esc-i	Shift-F6, Esc-i
PF19/F19	Shift-F7, Esc-o	Shift-F7, Esc-o
PF20/F20	Shift-F8, Esc-p	Shift-F8, Esc-p
PF21/F21	Shift-F9, Esc-a, Ctrl-F5	Shift-F9, Esc-a, Ctrl-F5
PF22/F22	Shift-F10, Esc-s, Ctrl-F6	Shift-F10, Esc-s, Ctrl-F6
PF23/F23	Shift-F11, Esc-d, Ctrl-F7	Shift-F11, Esc-d, Ctrl-F7
PF24/F24	Shift-F12, Esc-f, Ctrl-F8	Shift-F12, Esc-f, Ctrl-F8

## Key Map File Structure

The contents of the file will consist of lines that assign a 3270, 5250 or telnet function to a keyboard key sequence as follows:

```
<emulation function>=<keyboard key sequence>
```

The 3270, 5250 and telnet keys will each have their own section in the file. The 3270 section will be denoted by:

```
[3270]
```

The 5250 section will be denoted by:

```
[5250]
```

The telnet section will be denoted by:

```
[telnet]
```

A comment is started by the '#' character:

```
# This is a comment
```



## TN3270 Emulation

The following 3270 emulation functions may be mapped:

<b>Emulation Function</b>	<b>Name</b>
Attention	attention
Backtab	backtab
Cent Sign	cent_sign
Clear	clear
Cursor Select	cursor_select
Cursor Left	cursor_left
Cursor Right	cursor_right
Cursor Up	cursor_up
Cursor Down	cursor_down
Delete	delete
Device Cancel	device_cancel
DUP	duplicate
Home	home
Enter	enter
Erase EOF	erase_eof
Erase Input	erase_input
Field Mark	field_mark

<b>Emulation Function</b>	<b>Name</b>
Forward Tab	tab
Insert	insert
Newline	newline
PA1	pa1
PA2	pa2
PA3	pa3
Reset	reset
SYSREQ	sysreq
PF1 through PF24	pf1 through pf24

## TN5250 Emulation

The following 5250 emulation functions may be mapped:

<b>Emulation Function</b>	<b>Name</b>
Attention	attention
Backtab	backtab
Cent Sign	cent_sign
Clear	clear
Cursor Select	cursor_select
Cursor Left	cursor_left
Cursor Right	cursor_right
Cursor Up	cursor_up
Cursor Down	cursor_down
Cursor Home	cursor_home
Delete	delete
Device Cancel	device_cancel
DUP	duplicate
Enter	enter
Erase EOF	erase_eof
Erase Input	erase_input
Field +	field_plus

<b>Emulation Function</b>	<b>Name</b>
Field -	field_minus
Forward Tab	tab
Help	help
Hex	hex
Home	home
Insert	insert
Newline	newline
PA1	pa1
PA2	pa2
PA3	pa3
Reset	reset
Roll Down	roll_down
Roll Up	roll_up
SYSREQ	sysreq
F1 through F24	f1 through f24

## Telnet Emulation

The following Telnet emulation functions may be mapped:

<b>Emulation Function</b>	<b>Name</b>
Cursor right	cursor_right
Cursor left	cursor_left
Cursor up	cursor_up
Cursor down	cursor_down
F1 through F4	f1 through f4

## Keyboard Sequences

The keyboard key sequences that may be assigned to an emulation function include control and function keys. Sequences starting with the ESCAPE key are also allowed. The following table defines these keys:

Keyboard Function	Definition
Control keys	^ followed by 1 or more characters
Function keys	F1 through F12
Shift	+ preceding a sequence
Escape key	\e
Home	home
End	end
Page Up	page_up
Page Down	page_down
Del	del
Enter	enter
Backspace	backspace
Cursor Up	up
Cursor Down	down
Cursor Left	left
Cursor Right	right
Scroll Lock	scroll_lock

Keyboard Function	Definition
Pause	pause
Right Hand Control	right_hand_control

## Control Keys

Control keys are defined by a ^ followed by a character sequence. The allowed values are:

- l keys
- 'a' through 'z'
- f1 through f12
- home
- end
- page\_up
- page\_down
- del
- enter
- backspace
- '[', ']', '^', '\_'
- up
- down
- left
- right

## Examples

```
attention=^a    # Assign Control-a as the 3270 Attn key
```

```
newline=^enter # Assign Control-Enter as the 3270 Newline key
```

## Escape Sequence

An escape sequence is the ESCAPE key followed by one or more keys.

## Example

```
pf1=\e\e1      # Assign escape,escape,1 as PF1
```

## Other Keys

The following are some examples of key sequences other than control keys or escape sequences:

```
pf1=f1          # Assign F1 as the 3270 PF1 key

home=home       # Assign Home as the 3270 Home key

erase_eof=end   # Assign End as the 3270 Erase EOF key

pf13+=f1        # Assign Shift-F1 as the 3270 PF13 key

pa1+=^a         # Assign Control-Shift-a as the 3270 PA1 key
```



## Application Keypad in Telnet Mode

In telnet mode, the numeric keypad can be in 2 different modes:

- Numeric
- Application

In application mode, different codes are sent to the host for the period, comma, zero and enter keys. On Windows, some of the keypad keys are mapped differently than what is on the key caps. The following diagrams show two typical keypad layouts.

### 101 Key Keyboard

Num Lock	/	*	-
7 Home	8 !	9 Pg Up	+
4 ..	5	6 Æ	
1 End	2 Ø	3 Pg Dn	Enter
0 Ins		. Del	

The keys on this keypad are re-mapped as follows:

- Plus ('+') - Same as Enter
- Minus ('-') - Comma (',')

## 84 Key Keyboard

Esc	Num Lock	Scroll Lock	Sys Req
7 Home	8 	9 Pg Up	Prt Sc *
4 ..	5	6 Æ	-
1 End	2 Ø	3 Pg Dn	+
0 Ins		. Del	

The keys on this keypad are re-mapped as follows:

- Plus ('+') - Enter
- Minus ('-') - Comma (',')

## The Terminfo File

JI Integration requires that information in your terminfo file must be set correctly. It is possible that information in this file does not match the information sent by your keyboard, in which case JI Integration terminal emulation will not work correctly. Follow these directions to determine if the information is set correctly and to edit the file, if necessary.

- 1 Enter `od -c` at the command prompt and press the AID keys (F10, F11, F12, etc.). The escape sequence representing these keys will be echoed to the screen. Write down the escape sequences.
- 2 Next, enter `infocmp xterm | more` at the command prompt, which will print the terminfo information for x terminals. Verify that the escape sequences listed in the file for the AID keys (kf10, kf11, kf12, etc.) are defined correctly.
- 3 If they are not defined correctly, or are not defined at all, follow the remaining directions. Enter `infocmp xterm > infocmp.out` at the command prompt. Execute this command in a directory that you have set up specifically for containing your new terminfo file. Edit the *infocmp.out* file (which was just

created) so that the AID key entries match the actual escape sequences sent when the key is pressed (determined in <Hyper>Step 1, above).

- 4 Set the TERMINFO environment variable to point to the directory that holds `infocmp.out` by entering `export TERMINFO=<directory>` at the command prompt, where `<directory>` is the directory where `infocmp.out` was created.
- 5 Enter `tic infocmp.out` at the command prompt. This will generate new compiled terminfo entries in the directory to which the TERMINFO environment variable points.
- 6 Enter `infocmp xterm | more` again and verify that the new compiled terminfo is set correctly.
- 7 Stop and restart your JI Integration environment, and verify that the AID keys now work correctly. As long as TERMINFO is set, your newly compiled terminfo will be used.



## Chapter 7. Configuration Files

---

Configuration files are used to store values for configurable parameters associated with the various components of JI Integration. These configuration files are in ASCII text format and can be edited by any text editor. Parameters in the configuration files are assigned values using the syntax `name=value`, with no quotation marks or spaces included in the `name=value` pair. Blank lines are ignored, and all text following a number sign (#) is treated as a comment.

The default location for the configuration files is `<JI_install_dir>/config` where `<JI_install_dir>` represents the JI Integration installation directory. However, the configuration files can be stored anywhere on your file system, as long as the correct location is passed to the Environment Manager, Resource Server, and Configuration Manager when these applications are started (using the `-f` option). Similarly, the configuration files for the Environment Manager and Resource Server can be given any name as long as the correct name is passed to the application at startup. For more information about starting these applications, see the command line executables information in the *JI Integration User's Guide*.

The following table lists the JI Integration configuration files and describes the types of parameters defined in the file:

Configuration File	Description
<code>&lt;envmgr.cfg&gt;</code>	The Environment Manager configuration file.
<code>&lt;ressvr.cfg&gt;</code>	The Resource Server configuration file.
<code>&lt;cfgmgr.cfg&gt;</code>	The Configuration Manager configuration file.
<code>&lt;sysmon.cfg&gt;</code>	The System Monitor configuration file.
<code>environment.ccf</code>	Contains general parameters used to configure the your JI Integration environment.

**Note:** The “<>” syntax indicates the user may choose to rename this file.

## Order of Precedence for Command Line/Configuration File Parameters

Many of the items included in the configuration files can be passed as command line options when the component is started. This includes a number of parameters for the following components of the JI Integration system:

- Environment Manager
- Resource Server
- Configuration Manager

In all cases where there are multiple ways to pass configuration options to the application, JI Integration will process these options in the following order of precedence:

- 1 Command line options.
- 2 Configuration file parameters.
- 3 Database configuration records.
- 4 Hard coded default options.

For example, if an Environment Manager configuration file has the parameter `resourcename=ea` but the Environment Manager is launched with a command line option of `-r newresource` the resource used by the Environment Manager will be `newresource`, rather than `ea`.

## Windows Directory Names

In these configuration files, when using Windows directory names, all backslashes should be “escaped” with another backslash. Therefore, the following path:

```
c:\maps\gen\ji
```

should be written:

```
c:\\maps\\gen\\ji
```

When drive letters are not used, forward slashes can be used in place of drive letters. For example, the windows directory:

```
..\maps\gen\ji
```

could be written:

```
../maps/gen/ji
```

or

```
..\\maps\\gen\\ji
```

## The Environment Manager Configuration File

The Environment Manager configuration file is used to define parameters for the Environment Manager. The default name for this file is *envmgr.cfg* but any name can be used as long as the correct name is passed to the Environment Manager when the application is started (using the *-f* option).

**Note:** If you use the *ea\_start* and *ea\_shutdown* executables, along with other administrative executables, you should use the default name for this file.

The following is a list of the parameters included in the Environment Manager configuration file, along with a description of each parameter.

Parameter	Description
<code>configname*</code>	<p>The name of the Environment Manager configuration as it appears in the Resource Database. This parameter must be set to an existing configuration in the Resource Database or the Environment Manager will fail to start.</p> <p>Default: <code>eaconfig</code></p>
<code>envMCastAddr</code>	<p>The Environment Manager multicasting address; for example:</p> <p><code>ALL-SYSTEMS.MCAST.NET:35998</code></p> <p>The port number in this parameter must match the port number listed in the <code>mcast.listenaddr.env</code> parameter of the Resource Server's configuration file.</p>
<code>envMCastCacheTime</code>	<p>The cache time for the Environment Manager's multicasting, in milliseconds.</p> <p>-1 represents no caching.</p> <p>Default: -1</p>

**Note:** Parameters designated with an asterisk (\*) may be also be set via the JI Integration Configuration Manager.



Parameter	Description
envMCastTimeout	<p>The timeout value for the Environment Manager's multicasting, in milliseconds. -1 represents no timeout.</p> <p>Default: 5000</p>
lbMCastAddr	<p>The Environment Manager Load Balance Group multicast address. May be specified as:</p> <p>lbMCastAddr=[&lt;host&gt;][:&lt;port&gt;]</p> <p>Leaving the port unspecified allows newly created load balance groups to be started on any unused port.</p> <p>This property is set to: 239.192.0.0 by the JI Integration installer.</p> <p>Default ALL-SYSTEMS.MCAST.NET</p>
file	<p>The name and location of the Environment Manager's configuration file.</p>
groupname	<p>Sets the load balancing group name. When the Environment Manager is launched, it uses this parameter to either:</p> <ul style="list-style-type: none"><li>• Create a group using the value of this parameter as the group name, if that group does not already exist, or;</li><li>• Join an existing group, if that group has already been created by another Environment Manager.</li></ul> <p>All Environment Managers that are members of the same group must have the same value for groupname in their configuration files.</p> <p>Default: eagroup</p>
installdir	<p>The directory in which JI Integration was installed (for example, <i>/opt/JI</i> or <i>c:\JI</i>).</p>

**Note:** Parameters designated with an asterisk (\*) may be also be set via the JI Integration Configuration Manager.

Parameter	Description
<code>javacmd*</code>	The path to the Java Virtual Machine (JVM) that the Environment Manager will use. If the Java Runtime Environment (JRE) provided with JI Integration installation was installed and is being used, this property will default to <code>../jre/bin/java</code>
<code>jcclass</code>	The class name for the JI Integration JCluster class.  Default: <code>com.jacada.ea.jcluster.JCluster</code>
<code>license*</code>	The License Name used by this Environment Manager.  Default: CNT
<code>logfile*</code>	The name of the Environment Manager's log file. Optionally, this parameter can also set the path to the log file. By default, the location <code>&lt;JI_install_dir&gt;/logs</code> .  Additional valid entries are: <ul style="list-style-type: none"><li>• <code>System.out</code></li><li>• <code>System.err</code></li></ul> If no value is entered for this parameter, and the value of the <code>loglevel</code> parameter is greater than 0, logging is sent to <code>system.out</code> .  Default: <code>envmgr.log</code>
<code>loglevel*</code>	The level that information is logged to the Environment Manager's log file. Valid entries are 0 through 9, with 0 representing no logging and 9 representing the most verbose logging. Setting the log level to 0 turns logging off.  Default: 0

**Note:** Parameters designated with an asterisk (\*) may be also be set via the JI Integration Configuration Manager.

Parameter	Description
<code>nospawn</code>	Instructs the Environment Manager to not spawn any JClusters when it starts.
<code>resourceMCastAddr</code>	<p>The resource location multicast address, for example:</p> <p><code>ALL-SYSTEMS.MCAST.NET:35999</code></p> <p>The port number in this parameter must match the port number listed in the <code>mcast.listenaddr.res</code> parameter of the Resource Server's configuration file.</p>
<code>resourceMCastCacheTime</code>	<p>The cache time for the multicasting Resource location, in milliseconds. -1 represents no caching.</p> <p>Default: -1</p>
<code>resourceMCastTimeout</code>	<p>The timeout value for the multicasting resource location, in milliseconds. -1 represents no timeout.</p> <p>Default: 1000</p>
<code>resourcename*</code>	<p>The name of the Resource Database that contains configuration information for the Environment Manager. The location of the database is defined in the Configuration Manager; therefore, the value of this parameter must correspond to the name of the appropriate resource listed in the Configuration Manager.</p> <p>Default: <code>eardb://ea</code></p>
<code>rmilogging</code>	<p>If set to <code>true</code>, logs RMI debugging information to the log file specified in the <code>logfile</code> parameter.</p> <p>Default: <code>false</code></p>

**Note:** Parameters designated with an asterisk (\*) may be also be set via the JI Integration Configuration Manager.

Parameter	Description
rmiport	<p>The port number of the Remote Method Invocation (RMI) port, used to allow interaction between Environment Managers and Resource Servers. If the Resource Server and Environment Manager are running on the same machine, the value of this port can match the value for the <code>config.rmiport</code> parameter in the Resource Server configuration file or the <code>-p rmiport</code> command line option for the Resource Server. However, if the same RMI port is used for the Environment Manager and the Resource Server and the Resource Server goes down, the Environment Manager will have to be restarted.</p> <p>Default: 30002</p>
serverport	<p>The port number on which the Environment Manager listens for client connections.</p> <p>Default: 30001</p>
XMLEncoding	<p>Specifies the XML character coding to be used for the data returned from a method invocation. Possible XML character encoding values are specified by the W3 group and may be found at <a href="http://iana.org/assignments/character-sets">http://iana.org/assignments/character-sets</a></p> <p>Unless this parameter is specifically set, no character encoding will be specified.</p> <p>Default: NULL</p>

**Note:** Parameters designated with an asterisk (\*) may be also be set via the JI Integration Configuration Manager.

## The Resource Server Configuration File

The Resource Server configuration file is used to define parameters for the Resource Server. The default name for this file is *ressvr.cfg*, but any name can be used as long as the correct name is passed to the Resource Server when the application is started (using the *-f* option).

**Note:** If you use the *ea\_start* and *ea\_shutdown* executables, along with other administrative executables, you should use the default name for this file.

The following is a list of the parameters included in the Resource Server configuration file, along with a description of each parameter.

Parameter	Description
<code>config.installdir</code>	The directory in which JI Integration was installed (for example, <i>/opt/EA</i> or <i>c:\EA</i> ).
<code>config.logfile</code>	<p>The name of the Resource Server's log file. Optionally, this parameter can also set the path to the log file. By default, the path is <i>&lt;JI_install_dir&gt;/logs</i>.</p> <p>Additional valid entries are:</p> <ul style="list-style-type: none"> <li>• <i>System.out</i></li> <li>• <i>System.err</i></li> </ul> <p>If no value is entered for this parameter, and the value of the <i>loglevel</i> parameter is greater than 0, logging is sent to <i>system.out</i> by default.</p> <p>Default: <i>ressvr.log</i></p>
<code>config.loglevel</code>	<p>The level that information is logged to the logfile. Valid entries are 0 through 9, with 0 representing no logging and 9 representing the most verbose logging. Setting the <i>loglevel</i> to 0 turns logging off.</p> <p>Default: 0</p>
<code>config.maxlogbuffer</code>	The number of lines that the Resource Server will buffer before writing to the log file.

Parameter	Description
<code>config.rmilogging</code>	<p>If set to <code>true</code>, logs RMI debugging information to the log file specified in the <code>logfile</code> parameter.</p> <p>Default: <code>false</code></p>
<code>config.rmiport</code>	<p>The port name or number of the Remote Method Invocation (RMI) port, used to communicate between the Environment Manager and the Resource Server. If the Resource Server and Environment Manager are running on the same machine, the value of this port can match the value for the <code>rmiport</code> parameter in the Environment Manager's configuration file or the <code>-d rmiport</code> command line option for the Environment Manager. However, if the same RMI port is used for the Environment Manager and the Resource Server and the Resource Server goes down, the Environment Manager will have to be restarted.</p> <p>Default: <code>30002</code></p>
<code>db.driver_options</code>	<p>Additional options for JDBC driver.</p>
<code>db.ea_username</code>	<p>Default database user name. This parameter is only used if a database user name is required but is not provided by when the database is opened.</p>
<code>db.jdbc_driver</code>	<p>Fully qualified class name of JDBC driver to use.</p>
<code>db.mapfile</code>	<p>The name of the object map file, which maps objects to the database. Optionally, this parameter can also set the path to the object map file. By default, the location is <code>&lt;JI_install_dir&gt;/etc</code>.</p> <p>Default: <code>obj.map</code></p>

Parameter	Description
<code>db.primary_ea_database_url</code>	<p>If no database is specified when the database is opened, the value of this parameter will be used. This can be the name of a resource as listed in the resource map file, or the URL of a database, using the format <code>jdbc:h2:tcp://&lt;host&gt;:&lt;port&gt;/&lt;db_name&gt;</code>, where <code>&lt;host&gt;</code> is the name or IP address and <code>&lt;port&gt;</code> is the port name or number where the Resource Database server is running, and <code>&lt;db_name&gt;</code> is the name of the database.</p>
<code>db.url_header</code>	JDBC URL header for the specified driver.
<code>mcast.alt_bindto</code>	<p>Used by “satellite” proxy servers. The value of this parameters is a list of comma-separated <code>host:port</code> combinations, where <code>host</code> is the name or IP address and <code>port</code> is the port of the alternative hub proxy server. For example, given the following parameter:</p> <pre>mcast.alt_bindto=host1:3002, =host2:4002</pre> <p>The satellite server will first attempt to connect to its primary hub server defined in the <code>mcast.bindto</code> parameter. If that server does not respond, the satellite server will attempt to connect to the Resource Server running on <code>host1:3002</code>, then if that host does not respond, it will attempt to connect to <code>host2:4002</code>.</p> <p>Default: <code>null</code></p>
<code>mcast.bindto</code>	<p>Used by “satellite” proxy servers. The value of this parameters is the host name or IP address and port of the hub proxy server.</p> <p>Default: <code>null</code></p>

Parameter	Description
<code>mcast.poll_interval</code>	<p>Used by “satellite” proxy servers. If the hub is not active when the Resource Server is started, this is the amount of time to wait between polls of the hub. This value is in seconds.</p> <p>Default: 30 seconds</p>
<code>mcast.listenaddr.name</code>	<p>For advanced configuration purposes, this parameter allows you to set the multicast address and port on which the resource server or environment manager will listen. The Proxy Server will rebroadcast this address to its bind to address (via RMI) and to any other RMI servers which are using this server as a hub. There can be any number of these entries although <i>name</i> must be unique.</p> <p>Examples:</p> <pre>Mcast.listenaddr.env=   ALL-SYSTEMS.MCAST.NET:35998</pre> <pre>Mcast.listendaddr.res=   ALL-SYSTEMS.MCAST.NET:35999</pre> <p>The <code>mcast.listenaddr.res</code> parameter must have the same port number as the port specified for the <code>res.mcastport</code> parameter, and must also match the port specified for the <code>resourceMCastAddr</code> parameter in the Environment Manager’s configuration file.</p> <p>The <code>mcast.listenaddr.env</code> parameter must have the same port number as the port specified for the <code>envMCastAddr</code> parameter in the Environment Manager’s configuration file.</p>
<code>mcast.timeout</code>	<p>The amount of time the Resource Server will wait for a response to a multicast request in milliseconds. The minimum allowed value is 1000.</p> <p>Default: 100.</p>



Parameter	Description
<code>res.mapfile</code>	<p>The name of the resource map file. Optionally, this parameter can also set the path to the resource map file. By default, the location <code>&lt;JI_install_dir&gt;/config</code>.</p> <p>Default: <code>resource.map</code></p>
<code>res.mcastport</code>	<p>The Resource Server's resource location multicast port. Must have the same port number as the port specified for the <code>mcast.listenaddr.res</code> parameter, and must also match the port specified for the <code>resourceMCastAddr</code> parameter in the Environment Manager's configuration file.</p>
<code>service.alias.*</code>	<p>Defines alias for a service class. The format of the value is <code>&lt;alias name&gt;, &lt;fully qualified class name&gt;</code>. For example, to create an alias name <code>mcast</code> for the Proxy Server capabilities of the Resource Server, this parameter should be set to:</p> <pre>service.alias.mcast=mcast,com.jacada.ea.resserver.service.MCastProxyService</pre>
<code>service.max</code>	<p>The maximum number of services that may be loaded without client connections. Services will be launched in excess of this number when necessary to satisfy client connections, but any services in excess of this number will be killed when the <code>service.timeout</code> value is reached.</p>

Parameter	Description
<code>service.start.*</code>	<p>Starts the specified resource service on server initialization. The value can be either an alias defined in a <code>service.alias.*</code> parameter, or can be a fully qualified class name. For example, to enable the alias <code>mcast</code>, this parameter should be set to:</p> <pre>service.alias.mcast=mcast</pre> <p>or, to enable the Proxy Server capabilities of the Resource Server directly, this parameter could be set to:</p> <pre>service.start.mcast=com.jacada. ea.resserver.service. MCastProxyService</pre>
<code>service.timeout</code>	<p>The amount of time, in seconds, that a resource service instance will wait for a new client. Applies only to those resource services that are launched in excess of the <code>service.max</code> value.</p>

## The Configuration Manager Configuration File

The Configuration Manager configuration file is used to define parameters for the Configuration Manager. The default name for this file is *cfgmgr.cfg*, but any name can be used as long as the correct name is passed to the Configuration Manager when the application is started (using the *-f* option).

The following is a list of the parameters included in the Configuration Manager configuration file, along with a description of each parameter.

Parameter	Description
<code>autofind</code>	If set to <code>true</code> , instructs the Configuration Manager to use multicasting technology to automatically find all Resource Servers in your JI Integration environment.  Default: <code>true</code>
<code>IPAddrFormat</code>	If set to <code>True</code> , displays Resource Server and Environment Manager address in IP address format. If <code>False</code> , the DNS name will be displayed.  Default: <code>False</code>
<code>loglevel</code>	The level that information is logged to the logfile. Valid entries are 0 through 9, with 0 representing no logging and 9 representing the most verbose logging. Setting the <code>loglevel</code> to 0 turns logging off.  Default: 0
<code>lookandfeel</code>	Set the look and feel of the Resource Server GUI. Allowed values are <code>Windows</code> , <code>CDE/Motif</code> , <code>Metal</code> , <code>System</code> . If set to <code>system</code> , the GUI will use the default look and feel of the system on which the GUI is running.  Default: <code>System</code>  <b>Note:</b> The Windows “look and feel” is not available on UNIX.
<code>envMCastAddr</code>	Must match the <code>envMCastAddr</code> parameter in the Environment Manager’s configuration file.

Parameter	Description
<code>resourceMCastAddr</code>	Must match the <code>resourceMCastAddr</code> parameter in the Environment Manager's configuration file.

## The System Monitor Configuration File

The System Monitor configuration file is used to define parameters for the Configuration Manager. The default name for this file is *sysmon.cfg*, but any name can be used as long as the correct name is passed to the System Monitor when the application is started (using the `-f` option).

The following is a list of the parameters included in the System Monitor configuration file, along with a description of each parameter.

Parameter	Description
<code>autofind</code>	If set to <code>true</code> , instructs the System Monitor to use multicasting technology to automatically find all Resource Servers in your JI Integration environment.  Default: <code>true</code>
<code>IPAddrFormat</code>	If set to <code>True</code> , displays Resource Server and Environment Manager address in IP address format. If <code>False</code> , the DNS name will be displayed.  Default: <code>False</code>
<code>loglevel</code>	The level that information is logged to the logfile. Valid entries are 0 through 9, with 0 representing no logging and 9 representing the most verbose logging. Setting the <code>loglevel</code> to 0 turns logging off.  Default: 0
<code>lookandfeel</code>	Set the look and feel of the Resource Server GUI. Allowed values are <code>Windows</code> , <code>CDE/Motif</code> , <code>Metal</code> , <code>System</code> . If set to <code>system</code> , the GUI will use the default look and feel of the system on which the GUI is running.  Default: <code>System</code>  <b>Note:</b> The <code>Windows</code> “look and feel” is not available on UNIX.

## Environment Settings

### Environment Variables

Configuration files are used to define values for configurable parameters. However, the use of environment variables is supported for those parameters in the *environment.ccf* file only. To determine the value of a parameter, JI Integration will first search the *environment.ccf* configuration file for the parameter and its corresponding value. If the parameter does not exist or the value is not defined, JI Integration will then determine if an environment variable of the same name as the configurable parameter has been set. If the environment variable is found, JI Integration will then use the value of the variable.

For all other configuration files, JI Integration does not support environment variables as a replacement for the configuration parameters; for these parameters, you must set the value in the configuration file.

### EA\_ENV Environment Variable

There is one global environment variable associated with the JI Integration: EA\_ENV. EA\_ENV points to the location of your JI Integration configuration files. EA\_ENV can be set to a different location for each user, allowing each user to set parameters specific to their own environment.

## The `environment.ccf` Configuration File

The `environment.ccf` configuration file is used to set the following general JI Integration parameters.

Parameter	Description
DB_HOME	The directory location of the JI Resource Database installation. This property is not required when running the embedded H2 database.  <b>Deprecated.</b>
DB_PASSWORD	The password for the JI Resource Database server. This property can be used to set the password used by JI commands for access to the JI Resource Database server.
DB_PING_CMD	The command used by the JI command line utilities to ping the JI Resource Database server. This property can be used in conjunction with the DB_START_CMD and DB_SHUTDOWN_CMD properties to customize the JI environment startup and shutdown sequences. The default is to use the built-in <code>ea_dbping</code> command.
DB_SHUTDOWN_CMD	The command used by the JI command line utilities to shut down the JI Resource Database server. This property can be used to customize the JI environment shutdown sequence. The default is to use the built-in <code>ea_dbshutdown</code> command.
DB_START_CMD	The command used by the JI command line utilities to start the JI Resource Database server. This property can be used to customize the JI environment startup sequence. The default is to use the built-in <code>ea_dbsvr</code> command.
DB_TCP_PORT	Port number of the Resource Database server. The default is 30000.

Parameter	Description
DB_USER_ID	The User ID for the JI Resource Database server. This property can be used to set the User ID used by JI commands for access to the JI Resource Database server.
EA_ENVMGR	<p>Hostname and port number where the JI Integration Environment Manager is running, in the format</p> <pre>host:port [, host:port , host:port ...]</pre> <p>Where <code>host</code> and <code>port</code> can be defined as numbers or names. <code>host</code> can be the IP address or DNS name of the host. <code>port</code> can be the port number or a name. To provide for client load sharing, multiple Environment Managers may be defined, as shown in the example above.</p> <p>Default: <code>localhost:EA_Envmgr</code></p>
EA_HOME	Directory location of JI Integration installation.
PATERM_FIXED_SIZE	Determines whether the PA Terminal (JTerm) display size is fixed. Allowed values are <code>TRUE</code> (the PA Terminal window size is fixed) or <code>FALSE</code> (the PA Terminal window size is not fixed). The default is <code>FALSE</code> .
PATERM_FONT_SIZE	<p>The Protocol Agent Terminal initial font size. Valid sizes are: 6, 7, 8, 9, 10, 11, 12, 14, 16, 18, 20, 22, 24.</p> <p>Default: 7</p>
START_TOMCAT	Determines whether the Tomcat application server should be started with JI. Allowed values are <code>TRUE</code> (the Tomcat application server will be started after the Environment Manager) and <code>FALSE</code> (the Tomcat application server will not be started).



## Chapter 8. Logging Information

---

Throughout JI Integration, log files can be created to log debug information and to monitor the environment.

### Log File Location

The default location for JI Integration log files are located by default in the `<JI_install_dir>/logs` directory, where `<JI_install_dir>` represents the directory (and drive letter, if appropriate) where JI Integration was installed. This setting can be overridden, which is described for each component in the following sections.

### Using the System Monitor for Event Logging

The System Monitor can be used to log events that occur within the JI Integration environment. The events that are logged depend on what item is selected within the System Monitor. Also, the System Monitor is used to dynamically change log levels for JServices.

For more information about using the System Monitor, see Chapter 13: System Monitor of the *JI Integration User's Guide*.

## Environment Manager Logging

---

## Setting the Log Level

The Environment Manager log level controls the amount of information that is logged to the Environment Manager log file, as well as the amount of information that is logged to the JCluster(s) log file(s). The Environment Manager log level can be set in the following ways:

From within the Configuration Manager by connecting to the Environment Manager in the Server Node and changing the Log level parameter, which can be set to any number from 0-9. This method can be used to dynamically change the log level without having to shut down and restart the Environment Manager. For more information about configuring the Environment Manager, see “Configuring Environment Managers” in Chapter 2: Managing the JI Integration Environment of the *JI Integration User’s Guide*.

- At the command line when the Environment Manager is started, using the `-n` command line option (where *n* is any number from 0-9).
- In the Environment Manager’s configuration file, using the `loglevel` parameter, which can be set to any number from 0-9. For more information about the Environment Manager configuration file, see “Configuration Files” on page 131.

Using the command line option will override the setting in the Configuration Manager and the configuration file.

### Environment Manager Log Levels

The log level for the Environment Manager can be set to any value from 0 through 9, with 0 representing no logging and 9 representing the most verbose logging. Setting the `loglevel` to 0 turns logging off.

If logging is set to any non-zero number, information is logged to both the Environment Manager log file and log files for any JClusters that are controlled by the in the Environment Manager.

## Setting the Name and/or Location of the Log File

By default, the Environment Manager's log file is named *envmgr.log* and is located in the `<JI_install_dir>/logs` directory. However, both the name and location of the log file can be overridden in one of the following ways:

From within the Configuration Manager by connecting to the Environment Manager in the Server Node and changing the Log file parameter, which can be set to the name and/or path to the log file. For more information about configuring the Environment Manager, see "Configuring Environment Managers" in Chapter 2: Managing the JI Integration Environment of the *JI Integration User's Guide*.

- At the command line when the Environment Manager is started, using the `-l` command line option. The name and path to the logfile can be identified using this option. If no path is provided, the default location will be used. For more information about the Environment Manager command line interface, see "JI Integration Commands" on page 9.
- In the Environment Manager's configuration file, using the `loglevel` parameter. The name and path to the log file can be identified using this option. If no path is provided, the default location will be used. For more information about the Environment Manager configuration file, see "The Environment Manager Configuration File" on page 134.

Using the command line option will override the setting in the Configuration Manager and the configuration file.

## Information Logged to the Environment Manager Log File

The Environment Manager log file contains information about events, such as starting/stopping JClusters, multicasting events, and client connections. Also, any exceptions or other errors that are encountered by the Environment Manager are logged.

## JCluster Logging

---

JCluster logging is governed by the log setting for the Environment Manager. This setting can be changed dynamically in the Configuration Manager by connecting to the Environment Manager in the Server Node and changing the Log level parameter, which can be set to any number from 0-9. JCluster log levels can also be set at the command line when the Environment Manager is started, using the `-n` command line option (where `n` is any number from 0-9), or in the Environment Manager's configuration file, using the `loglevel` parameter, which can be set to any number from 0-9. These methods are not dynamic and require shutting down the Environment Manager and all JClusters before the changes take effect.

For more information about setting the log level for the Environment Manager, see "Environment Manager Logging" above.

## JCluster Log File Name

The name of the JCluster log file uses the following format:

`<IP address>-<port number>.log`

Where `<IP address>` is the address of the host machine on which the Environment Manager and JCluster(s) are running and `<port number>` is the port on which the JCluster is running. For example:

`255_93_35_13-2331.log`

## Information Logged to the JCluster Log File

The JCluster log file contains information about events, such as starting/stopping JServices, requests for services (RFSs) from the Environment Manager(s), and information about running the service. Also, any exceptions or other errors that are encountered by the JCluster are logged.

## JService Logging

---

## Setting the JService Log Level

The JService log level controls the amount of information that is logged to the JService log file, and can be set in the following ways:

- The trace level for the JService can be changed dynamically from within the System Monitor interface. To change the trace level, click in the Trace Level field on the JService Properties Panel. This parameter is an editable field that can be changed dynamically so that the amount of JService logging can be changed without having to shut down the JService or the JCluster. Allowed values are 0 to 9, with 0 indicating that there is no logging. For more information about the System Monitor, see Chapter 13: System Monitor of the *J1 Integration User's Guide*.
- The trace level can be set in the Configuration Manager in either the Service Master node in the Resources tree by setting the Default Trace Level parameter, or in the Service Detail node in the Resources tree (under the Environment Manager Configurations) by setting the Trace Level parameter. Both of these parameters can be set to any number from 0-9, and the Service Detail setting will override the Service Master default setting. For more information about using the Configuration Manager to create or edit Service Masters, see "Service Masters" in Chapter 12: Configuration Manager of the *J1 Integration User's Guide*. For information on creating or editing Service Details, see "Service Details" in Chapter 12: Configuration Manager of the *J1 Integration User's Guide*.

### JService Log Levels

The log level for JServices can be set to any value from 0 through 9, with 0 representing no logging and 9 representing the most verbose logging. Setting the loglevel to 0 turns logging off. If logging is set to any non-zero number, information is logged to the JService log file.

Setting the log level to 9 causes the full TN3270 trace to be logged to the JService log file.

## JService Log File Name

The name of the JService log file uses the following format:

```
service_<service name>_<service instance>_<port>.log
```

Where `<service name>` is the name of the Service Master/Service Detail as defined in the Configuration Manager, `<service instance>` is the sequential instance number of the service instance (displayed in parentheses in the System Monitor), and `<port>` is the port number of the JCluster that controls the JService.

For example:

```
service_EATutorialSvc_0_2331.log
```

## Dynamically Setting JCluster and JService Logging

---



## **Dynamically Changing the JCluster Trace Level**

The trace level for JClusters can be changed dynamically from within the Configuration Manager interface by connecting to the Environment Manager in the Server Node and changing the Log level parameter, which can be set to any number from 0-9. This method can be used to dynamically change the log level without having to shut down and restart the Environment Manager.

## Dynamically Changing the JService Trace Level

The trace level for JServices can be changed dynamically from within the System Monitor interface. To change the trace level, click in the Trace Level field on the JServices Properties Panel. This parameter is a editable field that can be changed dynamically so that the amount of JService logging can be changed without having to shut down the JService or the JCluster. Allowed values are 0 to 9, with 0 indicating that there is no logging. For more information, see “JService Events and Activities” in Chapter 13: System Monitor of the *J1 Integration User’s Guide*.

## Resource Server Logging

---

## Setting the Log Level

The Resource Server log level can be set in the following ways:

- At the command line when the Resource Server is started, using the `-n` command line option (where *n* is any number from 0-9).
- In the Resource Server's configuration file, using the `loglevel` parameter, which can be set to any number from 0-9.

Using the command line option will override the setting in the configuration file.

## Name of the Log File

By default, the Resource Server's log file is named *ressvr.log* and is located in the `<JI_install_dir>/logs` directory. However, both the name and location of the log file can be overridden in one of the following ways:

- At the command line when the Resource Server is started, using the `-l` command line option. The name and path to the log file can be identified using this option. If no path is provided, the default location will be used.
- In the Resource Server's configuration file, using the `logfile` parameter. The name and path to the log file can be identified using this option. If no path is provided, the default location will be used.

Using the command line option will override the setting in the configuration file.

## MapMaker Logging

---

MapMaker information is logged to the MapMaker log files, `mapmaker_stdout.txt` and `mapmaker_stderr.txt` located in the `<JI_install_dir>/logs` directory. Errors and events are logged to these files.

## Code Generation Log Files

If errors occur during code generation, a log file may be created in the same directory as the generated code.

## MapPlayer Logging

---

MapPlayer information is logged to the MapPlayer log files, `mapplayer_stdout.txt` and `mapplayer_stderr.txt` located in the `<JI_install_dir>/logs` directory. Errors and events are logged to these files.

## Resource Database Server Logging

---

Resource Databas server log messages are logged to the `dbsvr-<port>.log` file, where `<port>` is the TCP port number the Resource Database server is listening on.

## Configuration Manager Logging

---

Configuration Manager log file information is logged to the Configuration Manager log files, `cfgmgr_stdout.txt` and `cfgmgr_stderr.txt` located in the `<JI_install_dir>/logs` directory. Errors and events are logged to these files.

## System Monitor Logging

---

System Monitor information is logged to the System Monitor log files, `sysmon_stdout.txt` and `sysmon_stderr.txt` located in the `<JI_install_dir>/logs` directory. Errors and events are logged to these files.

## Service Logging

The name of the Service log file uses the following format:

`<service name>.<service pid>.<service instance>.log`

Where `<service name>` is the name of the Service Master/Service Detail as defined in the Configuration Manager, `<service pid>` is the process id of the service, and `<service instance>` is the sequential instance number of the service instance (displayed in parentheses in the System Monitor). For example:

`EATutorialSvc.45364.3.log`

## Chapter 9. National Language Support

---

JI Integration provides support for a number of single-byte and double-byte character sets. This allows a variety of languages to be used in conjunction with JI Integration services. The national language translation is performed by selecting the appropriate language when setting the host configuration in MapMaker. See the *JI Integration User's Guide* for more information about MapMaker and Java Services.

**Note:** Double-byte character sets are only supported for TN3270 and TN5250 in Java Services.

### Single-Byte Character Sets

---

The following single-byte character sets are supported:

- Windows Latin-1 (Cp1252)
- \*English(US) (Cp273)
- \*German (Cp037)
- \*Danish (Cp277)
- \*Finnish (Cp278)
- \*Italian (Cp280)
- \*Spanish (Cp284)
- \*English(UK) (Cp285)
- \*French (Cp297)
- Arabic (Cp420)
- Hebrew (Cp424)
- \*Belgian (Cp500)
- PC Greek (Cp737)
- PC Baltic (Cp775)
- Thai extended SBCS (Cp838)
- Hebrew (Cp856)
- Cyrillic (Cp855)
- Turkish (Cp857)
- Modern Greek (Cp869)
- Multilingual Latin-2 (Cp870)
- Icelandic (Cp871)

- \*Greek (Cp875)
- Estonia (AIX, DOS) (Cp922)
- \*Latin-5, Turkey (Cp1026)
- Estonia (Cp1122)
- \*English(US) with Euro (Cp1140)
- \*Finnish with Euro (Cp1143)
- \*English(UK) with Euro (Cp1146)
- Icelandic with Euro (Cp1149)
- Pakistan (Urdu) (Cp918)
- Pakistan (Urdu) (Cp1006)
- Farsi/Persian (Cp1097)
- Ukraine (Cp1123)
- \*German with Euro (Cp1141)
- \*Italian with Euro (Cp1144)
- \*French with Euro (Cp1147)
- Latvia, Lithuania (AIX, DOS) (Cp921)
- Multilingual Cyrillic (Cp1025)
- Latvia, Lithuania (Cp1112)
- AIX Ukraine (Cp1124)
- \*Danish with Euro (Cp1142)
- \*Spanish with Euro (Cp1145)
- \*Belgian with Euro (Cp1148)

**Note:** Languages marked with an asterisk (\*) have been formally tested with JI Integration. For information on support of all other languages listed, please contact Software GmbH Global Support.

## Double-byte Character Sets

---

Japanese is the only double-byte character set that has been certified with this release of JI Integration. However, the following character sets will be available with a later release of JI Integration:

- Thai (Cp874)
- Japanese Katakana-Kanji mixed 4370 UDC, superset of 5026 (Cp930)



- Korean mixed 1880 UDC, superset of 5029 (Cp933)
- Traditional Chinese mixed 6204 UDC, superset of 5033 (Cp937)
- Japanese, superset of Cp932 (Cp942)
- Japanese, superset of Cp932 and Shift-JIS (Cp943)
- AIX Chinese (Taiwan) (Cp964)
- People's Republic of China (PRC) (Cp1381)
- Japanese (superset of 5050) (Cp33722)
- Simplified Chinese mixed 1880 UDC, superset of 5031 (Cp935)
- Japanese Latin Kanji mixed 4370 UDC, superset of 5035 (Cp939)
- Japanese, superset of Cp932(Variant) (Cp942C)
- Japanese, superset of Cp932 and Shift-JIS(Variant) (Cp943C)
- AIX Korean (Cp970)
- People's Republic of China (PRC) (Cp1383)

**Note:** Please contact Software GmbH Global Support for updated information on double-byte character set support.



## Chapter 10. LAX Files

---

Each executable program provided with JI Integration includes a “LAX” file, which used to determine various properties that the program requires to execute. For example, JI Integration programs require a JVM (Java Virtual Machine) to execute. The LAX file that corresponds to the executable contains the information required to locate the JVM, which is included in the LAX file during JI Integration installation, based on the JVM selected in the installation program.

LAX files are created by the JI Integration installation program, which uses ZeroG’s InstallAnywhere™ Java installation. Included with this installation program is ZeroG’s LaunchAnywhere Java application launching technology. This technology uses LAX files to provide information to Java executables. Each LAX file will have the same name as its corresponding executable. For example, the *ea\_envmgr* executable has a corresponding lax file named *ea\_envmgr.lax*. JI Integration LAX files are located in the `<JI_install_dir>/bin` directory.

## Editing LAX Files

---

Occasionally, it may be necessary to edit LAX files. For example, during JI Integration installation, LAX files are created that contain information about the location of your Java Virtual Machine (JVM). However, after installing JI Integration, you might install a newer JVM or you may want to use a different JVM than the one you selected during installation. If this is the case, you will have to edit the LAX files included with JI Integration so that they point to the correct JVM.

LAX files are ASCII text files which can be edited by any text editor such as vi or Notepad.

### LAX File Content

LAX files consist of a number of sections, identified by the commented line `# LAX.<SECTION_NAME>`. For example, the section `LAX.CLASS.PATH` identifies the classpath required to launch the application. Each section contains name=value pairs indicating the properties included in the section, followed by the value for the property.

## Redirecting Standard Error and Standard Out

There are two sections in the LAX file for redirecting standard output from the application. These sections, `LAX.STDERR.REDIRECT` and `LAX.STDOUT.REDIRECT` allow for standard output to be written to a file.

### Creating No Standard Output

Additionally, these sections can also be used to turn standard output off. However, the comment in these sections states that the value can be left blank for no output. This is incorrect. To instruct the application to create no standard output, the value for these sections must be set to `NULL` for example:

```
lax.stdout.redirect=NULL
```

## Changing the JVM Used to Launch JI Integration Applications

The most common change that will be made to the LAX files included with JI Integration will be to change the Java Virtual Machine (JVM) used to launch JI Integration executables. For example, you may want to use a new JVM to run the applications. To change the JVM used by JI Integration applications, open the LAX file in a text editor such as vi or Notepad. Locate the section entitled `LAX.NL.CURRENT.VM`. The contents of this section will vary depending on what JVM was selected when JI Integration was installed, and also depending on whether JI Integration was installed on Windows or UNIX. The following are examples of this section, using the JVM that was included with JI Integration installation:

Windows example:

```
#   LAX.NL.CURRENT.VM
#   -----
#   the VM to use for the next launch

lax.nl.current.vm=D:\\EA\\jre\\bin\\java.exe
```

UNIX example:

```
#   LAX.NL.CURRENT.VM
#   -----
#   the VM to use for the next launch

lax.nl.current.vm=/jre142/jre/bin/java
```

To change the location of the JVM, change the value for the `lax.nl.current.vm` property to the appropriate location of your JVM. Note that in Windows, when drive letters are used, backslashes should be "escaped" using an additional backslash; therefore, each backslash should be written as two backslashes.

## Example LAX File

---

The following is an example of the Environment Manager's LAX file:

```
#   LaunchAnywhere (tm) Executable Properties File - Zero G Software, Inc.

java.security.manager=null

java.security.policy=C:\\JI\\etc\\.java.policy
```

```
# LAX.APPLICATION.NAME
# -----
# the default name of this executable -- do not edit

lax.application.name=ea_envmgr.exe


# LAX.CLASS.PATH
# -----
# the Java classpath necessary to run this application
# Can be separated by colons (Mac OS/Unix) or semicolons (Windows)

lax.class.path=C:\\JI\\classes\\common.jar;lax.jar


# LAX.COMMAND.LINE.ARGS
# -----
# what will be passed to the main method -- be sure to quote arguments with
# spaces in them

lax.command.line.args=-I C:\\JI $CMD_LINE_ARGUMENTS$


# LAX.DIR
# -----
# path to directory holding LaunchAnywhere's native launcher

lax.dir=C:\\JI\\bin\\


# LAX.MAIN.CLASS
# -----
# the class that contains the main method for the application

lax.main.class=com.jacada.ea.envmanager.app.EAServer


# LAX.MAIN.METHOD
# -----
# the method in the main class that will be invoked
```

```
lax.main.method=main

#   LAX.NL.CURRENT.VM
#   -----
#   the VM to use for the next launch

lax.nl.current.vm=C:\\j2sdk1.4.2_05\\bin\\java.exe

#   LAX.NL.JAVA.LAUNCHER.MAIN.CLASS
#   -----
#   main class of LaunchAnywhere's java launcher -- do not adjust

lax.nl.java.launcher.main.class=com.zerog.lax.LAX

#   LAX.NL.JAVA.LAUNCHER.MAIN.METHOD
#   -----
#   main method of LaunchAnywhere's java launcher -- do not adjust

lax.nl.java.launcher.main.method=main

lax.nl.java.option.additional=-Xmx256M

#   LAX.NL.VALID.VM.LIST
#   -----
#   a string containing one or more of [ ALL JDK JRE J1 J2 JRE_J1 JDK_J1
JRE_J2 JDK_J2 MSJ MRJ ]
#   delimited by spaces or commas.  If the native launcher cannot find the
current vm,
#   it will search for ones in this list

lax.nl.valid.vm.list=J2 J1 MSJ MRJ

#   LAX.NL.WIN32.MICROSOFTVM.MIN.VERSION
#   -----
```

```
# The minimum version of Microsoft's VM this application will run against

lax.nl.win32.microsoftvm.min.version=2750


# LAX.ROOT.INSTALL.DIR
# -----
# path to the installdir magic folder

lax.root.install.dir=C:\\JI


# LAX.STDERR.REDIRECT
# -----
# leave blank for no input, "console" to read from the console window,
# and any path to a file to read from that file

lax.stderr.redirect=C:\\JI/logs/envmgr_stderr.txt


# LAX.STDIN.REDIRECT
# -----
# leave blank for no input, "console" to read from the console window,
# and any path to a file to read from that file

lax.stdin.redirect=


# LAX.STDOUT.REDIRECT
# -----
# leave blank for no input, "console" to read from the console window,
# and any path to a file to read from that file

lax.stdout.redirect=C:\\JI/logs/envmgr_stdout.txt


# LAX.USER.DIR
# -----
# left blank, this property will cause the native launcher to not
# alter the platform default behavior for setting the user dir.
# To override this you may set this property to a relative or absolute
# path.
```



```
#   Relative paths are relative to the launcher.

lax.user.dir=..

#   LAX.VERSION
#   -----
#   version of LaunchAnywhere that created this properties file

lax.version=5.0
```



## Chapter 11. Licensing Procedures

---

This chapter describes:

- “The License Key” on page 178
- “License Properties” on page 179
- “The License File (license.txt)” on page 181
- “Adding and Editing a License Key” on page 182
- “Checking the Status of JI Integration Licenses” on page 185
- “License Expiration Email Notification” on page 187
- “Duplicate Licenses” on page 190
- “License Compatibility” on page 191

## The License Key

The JI Integration license key contains the following information:

- 1 License Name
- 2 Major Version
- 3 Minor Version
- 4 Expiration Date and time
- 5 One to five licenses — each license contains a combination of JI Integration features required by a specific customer. The available features include JI\_SERVER, JI\_SERVICE, JI\_CLIENT, JI\_HOST and JI\_MAPMAKER. The empty slots are indicated by the value NO\_LICENSE. For information on the various license features, see “License Properties” on page 179.

## License Properties

Each of the licenses included in the license key contains the following information:

- “Feature Type” on page 179
- “License Count” on page 180
- “Expire Flag” on page 180

### Feature Type

This property determines the maximum number of instances that a specific type of feature may have across *all* active JI Integration runtimes. The license feature type can have one of the following values:

- JI\_SERVER — determines the maximum number of servers on a network, permitted to run a JI Integration Environment Manager.
- JI\_SERVICE — determines the maximum number of concurrent Services.
- JI\_CLIENT — determines the maximum number of concurrent run-time clients that are external to the JI Integration Server and are connected to active services.

**Note:** Although JClient3 method steps are JI Integration Clients, they are not counted because they are “internal” clients within the JI Integration Server.

- JI\_HOST — determines the maximum number of concurrent host sessions.
- JI\_MAPMAKER — allows an unlimited number of MapMaker GUIs to run simultaneously.

In addition to these five standard features, two more feature types may be used:

- NO\_LICENSE — indicates that this license key slot has not been filled.
- JI\_DEMO — allows you to get acquainted with JI Integration through a composite of three features: JI\_SERVER (one license), JI\_MAPMAKER (unlimited licenses) and JI\_SERVICE (as many licenses as there are JI\_DEMO licenses, which range from one to five).

**Note:** The number of licenses allowed by JI\_DEMO is overridden by the number of licenses allowed by non-demo features.

### License Priority

JI\_SERVICE, JI\_CLIENT and JI\_HOST are not used concurrently. Instead, the license checking code searches for one valid license in the following order:

- 1 JI\_SERVICE
- 2 JI\_CLIENT
- 3 JI\_HOST

Once the code finds the license with the highest priority available, it stops searching for licenses that have a lower priority. The code uses the information in the license that takes precedence, and automatically sets the other two licenses to “unlimited”.

For example, if JI\_SERVICE is not defined and JI\_CLIENT and JI\_HOST are both defined, JI\_CLIENT has the highest priority available. Accordingly, the JI\_CLIENT information will be used while JI\_SERVICE and JI\_HOST will be set to “unlimited”.

### License Count

The license count determines how many instances of this feature may be used concurrently. This number is either a numeric value, or the string “unlimited”.

**Note:** JI Integration licenses are not counted per Server instance, but across all active JI Integration runtimes. For example, if you are running two JI Integration Servers and your license is for 100 Services, you can only run 100 total Services between the two Servers (and not 100 Services per-Server).

### Expire Flag

JI Integration licenses are renewed once a year. The expire flag indicates whether or not the license should honor the expiration date.

## The License File (license.txt)

The JI Integration licensing information is located in a special license file. The file's default name is `license.txt` and its default location is in the `<JI_install_dir>/config` directory.

The `license.txt` file is created the first time a license is entered during the installation or via the Configuration Manager. The file is served by a Resource Server and is edited using the Configuration Manager. The configuration manager always shows the same license keys as the `license.txt` file.

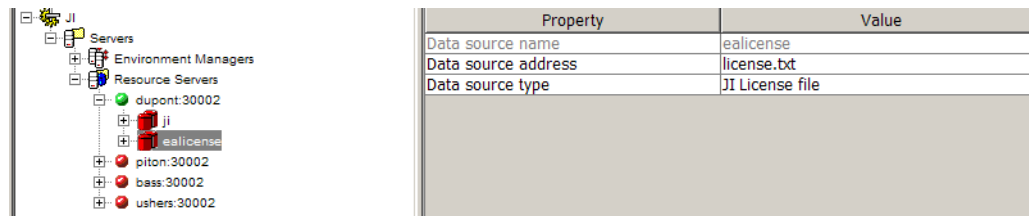
**Note:** Do not edit the `license.txt` file using any tool other than the Configuration Manager.

To edit or add license keys within the license file, you must first connect to the Resource Server (see the “Locating Resource Servers” section in Chapter 12: Configuration Manager of the *JI Integration User's Guide*).

**Note:** Licensing information also appears in the `envmgr.cfg` configuration file, which specifies the name of the license entered during the JI Integration installation. If several licenses have been added using the Configuration Manager, the license name specified in `envmgr.cfg` determines which license is used by that environment manager at runtime.

## Adding and Editing a License Key

After connecting to the Resource Server, a default license resource named `ealicense` is listed as subordinate to that Resource Server (Figure 1).



**Figure 1. Default License Resource — ealicense**

The Properties view displays the values of the following license resource properties:

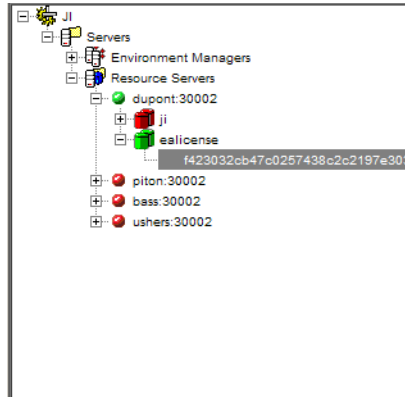
Property	Description
Data source name	The name of the resource. This value cannot be edited.
Data source address	<p>The location of the license file:</p> <p><code>&lt;directory&gt;/&lt;filename&gt;.txt</code></p> <p>Examples:</p> <ul style="list-style-type: none"> <li>UNIX:  <code>/opt/JI/config/license.txt</code></li> <li>Windows: <code>C:\JI\config\license.txt</code></li> </ul> <p><b>Note:</b> If the license file is in its default location <code>&lt;JI_install_dir&gt;/config</code> you do not need to enter the path to the file.</p>
Data source type	The type of data source identified by this resource.

To view the license properties, right click the license node in the Tree view and select **Open datasource** from the pop-up menu.

The license key nodes are displayed in the Tree view. Selecting one of the license nodes displays its properties in the Properties view (Figure 2).



**Note:** Open license data source nodes have a green icon; those that are closed have a red icon.



Property	Value
License key	f423032cb47c0257438c2c2197e3032403...
License name	SAG
Expiration date	2011-12-31 00:00:00
Major version	4
Minor version	5
0: Feature name	JI - Server License
0: Number of licenses	1
0: License expires?	No
1: Feature name	JI - Service License
1: Number of licenses	1
1: License expires?	No
2: Feature name	JI - MapMaker License
2: Number of licenses	1
2: License expires?	Yes
3: Feature name	JI - Client License
3: Number of licenses	1
3: License expires?	No
4: Feature name	JI - Host License

**Figure 2. License Key Node Properties**

If the **License name** is something other than the default name (**SoftwareAG**) or **EmergencyTemporaryKey**, it is necessary to change the license parameter for the Environment Manager to match the License name entered here.

**Note:** A License name is not configurable, as it is associated with the Software GmbH-provided license key.

If the Environment Manager is not running, the Configuration Manager parameters for the Environment Manager are not editable. The value of the License name parameter needs to be changed directly in the Environment Manager's configuration file. For information about editing this file, see Chapter 7 - "Configuration Files" on page 131.

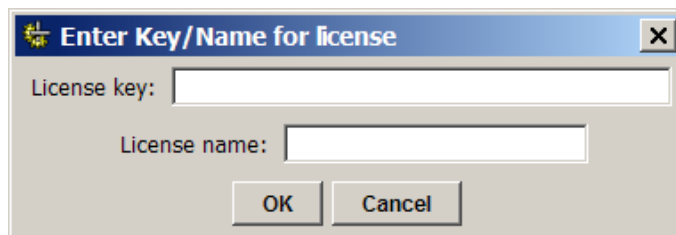
If the License Key is invalid, the Configuration Manager displays an error message and the License Key is rejected.

## Adding an Additional License Key Node

To add an additional license key node, proceed as follows:

- 1 Right-click the license node in the Tree view, and select **New License Key Node** from the pop-up menu.

The **Enter Key/Name for license** dialog box (Figure 3) opens.



**Figure 3. The License Key dialog box**

- 2 Enter the unique **License Key** and **License Name** that were provided with the installation or by a JI Integration sales or support representative.
- 3 Click on the **OK** button. The key's properties are listed in the Properties view (Figure 2). These properties are not editable.

## Adding an Additional License Data Source

To add a license datasource to a Resource Server, proceed as follows:

- 4 Right-click on the Resource Server node and select **New Data Source** from the pop-up menu
- 5 Enter a name for the datasource (for example, `licenseNew`).
- 6 In the Properties view, enter the following license details:
  - Specify the name of the license in the **Data Source Name** field.
  - Specify the license location in the **Data source address** field.
  - Select **JI License file** as the license type from the **Data source type** drop-down list.
- 7 Right-click and select **Save** from the popup menu (or select **Save** from the **File** pull-down menu).
- 8 To add License Keys, see “Adding and Editing a License Key” on page 182.

**Note:** If a license data source is added, the Resource Server must be restarted before the change takes effect.

## Checking the Status of JI Integration Licenses

The following table describes useful commands for checking license statuses.

Command	Function	Comment
<code>ea_start</code>	Checks for expired licenses.	<p>When the <code>ea_start</code> command starts the Resource Server, it also checks for licenses that have either expired or are about to expire in 30 days or less (and specifies the number of days or hours left).</p> <p><b>Note:</b> <code>ea_start</code> always shows the expiration status of the JI_DEMO license (e.g. "WARNING: Feature JI_DEMO, name PHOTON expires in 149 days.").</p>
<code>ea_admin_status</code>	Counts Service, Host and Client Licenses	<ul style="list-style-type: none"> <li>When checking for available licenses, make sure that no JI Integration Services have been pre-started. To identify the names of such pre-started Services, run the command <code>ea_admin jclusters -e</code></li> <li>For <code>ea_admin status</code> to display correct information after deleting or adding licenses using the configuration manager, you must restart the environment manager.</li> </ul>
<code>ea_admin licenseStatus</code>	Lists all licenses in the Resource Server.	<p>The licenses are listed in the following format:</p> <pre>&lt;feature Name&gt;&lt;License Name&gt; &lt;Expiration&gt;&lt;Count&gt;</pre> <ul style="list-style-type: none"> <li>Expiration: the number of hours until expiration, or "N" if the license never expires.</li> <li>Count: the number of licenses that can run simultaneously, or "unlimited" if there is no limit.</li> </ul>

Command	Function	Comment
<code>ea_admin jclusters -e</code>	Describes all JClusters and the services they contain.	

## License Expiration Email Notification

The Resource Server can be configured to send a daily email, listing licenses that are about to expire in 30 days or less. This section describes the email notification's format, and the email properties you are required to define in the Resource Server Configuration File (*ressvr.cfg*).

### Email Format

This license expiration email notification appears in the following format:

Email Field	Contents
Subject	"JI Integration License Expiration"
From Address	The email sender's address can be one of the following: <ul style="list-style-type: none"><li>• If the <b>config.fromUser</b> value is defined in the <i>ressvr.cfg</i> file, it appears in the <b>From</b> field.</li><li>• If the <b>config.fromUser</b> value is not defined, &lt;user_name&gt;@&lt;IP_Address&gt; is used instead, where:<ul style="list-style-type: none"><li>• &lt;user_name&gt; is the name of the user that started the Resource Server.</li><li>• &lt;IP_Address&gt; is the numerical IP address of the machine on which the Resource Server is running.</li></ul></li></ul>
From Name	"Resource Server"

Email Field	Contents
Body	<p>The email body consists of the following elements:</p> <ul style="list-style-type: none"><li>• “Resource Server Location: &lt;location&gt;”, where &lt;location&gt; is the host and port of the resource server in a &lt;host&gt;:&lt;port&gt; format.</li><li>• “Resource Server Version: &lt;version&gt;”.</li><li>• The following information is provided for each license that expires in 30 days or less:<ul style="list-style-type: none"><li>• “Feature &lt;feature name&gt;, name &lt;license name&gt; expires in N days”, Or</li><li>• “Feature &lt;feature name&gt;, name &lt;license name&gt; is expired”.</li></ul></li><li>• The following information is provided for each license that has already expired:<ul style="list-style-type: none"><li>• If &lt;feature name&gt; is JI_DEMO, the text “Your JI Integration server will stop functioning” is appended.</li><li>• If &lt;feature name&gt; is JI_MAPMAKER, the text “MapMaker will switch into Read Only mode. This will not affect your JI Integration server.” is appended.</li></ul></li><li>• The license key itself is also appended to the message, along with a breakdown of the key’s contents.</li></ul>
To Address, To Name	<p>The email recipient’s address and name are configured in <i>ressvr.cfg</i> using the following parameters:</p> <ul style="list-style-type: none"><li>• Recipient address: <b>config.mailHost</b></li><li>• Recipient name: <b>config.mailUser</b>.</li></ul> <p><b>Note:</b> To enable email notification, both <b>config.mailHost</b> and <b>config.mailUser</b> must be defined</p>

## Resource Server Configuration File (*ressvr.cfg*) Email Definitions

To following license expiration email properties must be defined in the Resource Server Configuration File (*ressvr.cfg*):

Parameter	Description
config.mailHost	<p>The host name or IP address of the machine to which email should be sent. If the mail server is not listening on the default SMTP port (25), the format "&lt;host&gt;:&lt;port&gt;" may be used.</p> <p><b>Note:</b> The mail host may not be the same as the company name.</p>
config.mailUser	<p>The email address (e.g. john_smith@mycompany.com). This value may contain multiple destination addresses, separated by commas.</p>
config.fromUser	<p>The email sender's address (e.g. john_smith@mycompany.com). If this value is not configured, a default value of &lt;user_name&gt;@&lt;IP_Address&gt; is used, where</p> <ul style="list-style-type: none"> <li>• &lt;user_name&gt; is the name of the user that started the Resource Server.</li> <li>• &lt;IP_Address&gt; is the numerical IP address of the machine on which the Resource Server is running.</li> </ul>
config.licenseCheckTime	<p>The hour at which the Resource Server checks for impending license expirations. The valid values are 0 through 23 (0 = midnight, 23 = 11:00 PM).</p> <p><b>Note:</b> This parameter is optional: if it is not defined, a default value of 7 (7:00 AM) is used.</p>

## Duplicate Licenses

The exact same license, with the same name and key, cannot be added more than once into the configuration manager. However, the configuration manager allows you to add multiple licenses that share the same name as long as they do not have an identical key. Such licenses are known as duplicate licenses.

If duplicate licenses have duplicate features (for example, two licenses with the same name both have a `JI_SERVER` feature), the Resource Server uses the first feature listed in the `license.txt` file and deletes all duplicate features following it.

**Note:** If a license displayed in the Configuration Manager has no features, it means that all of its features were duplicates deleted by the Resource Server.

The configuration manager always adds new licenses at the beginning of the list, so that if there are duplicate features the most updated license will take precedence.



## License Compatibility

JI Integration versions 4.X and 3.X can run on the same sub net, with the following limitations:

- 1** A 4.X Resource Server and a 4.X Configuration Manager utility cannot use 3.X licenses.
- 2** A 3.X Resource Server and a 3.X Configuration Manager utility cannot use 4.X licenses.
- 3** A 4.X Resource Server cannot serve licenses to 3.X environments.
- 4** A 3.X Resource Server cannot serve licenses to 4.X environments.



## Chapter 12. Uninstalling JI Integration Software

---

The JI Integration installation program creates an uninstall program to simplify removal of JI Integration software. The uninstall program will remove everything that was installed into the JI Integration installation directory, except for configuration information such as files in the `<JI_install_dir>/config` directory, and database files in the `<JI_install_dir>/databases` directory. Additionally, any files that are created in the `<JI_install_dir>` hierarchy after installation will cause the parent directory structure and the new files themselves to not be removed by the uninstaller.

This means that any changes to batch files, scripts, and `.lax` files will not be saved when the JI Integration software is uninstalled. If you intend to reinstall JI Integration and wish to use these files, you should back up any changed files and databases prior to uninstalling the JI Integration software.

Follow these directions to uninstall the JI Integration software:

### Windows:

- 1 Select Start Menu—Settings—Control Panel to open the Windows Control Panel.
- 2 Double-click on the Add/Remove Programs icon in the Control Panel. This will open the Add/Remove Programs applet.
- 3 Highlight the JI Integration entry in the list box and select the **Add/Remove** button to remove the JI Integration software.

### UNIX:

- 1 At the command prompt, change to the `<JI_install_dir>/UninstallerData<ea_version>` directory, where `<ea_version>` represents the version number of your JI Integration installation.

- 2 Enter the following command:

```
./Uninstall
```

This will remove the JI Integration software.

An additional file, `.mapmaker.cfg` is installed during JI Integration installation. This file contains configuration information related to MapMaker and is not editable. This file is not removed by the uninstall program, and is located in the root of the user's home directory in UNIX or the `c:\Windows\Profiles\<username>[.xxx]` directory in Windows (where `<username>` represents the user's login name and `[xxx]` represents the optional extension that may be created when the user logs on at different times using different domains).



## Chapter 13. Field Attribute Definitions

---

### IBM 3270 Field Attribute Byte

---

When an JI Integration service sets the ATTRIBUTES Protocol Agent option to ON for the TN3270 Protocol Agent, then IBM 3270 field attribute characters in the presentation space will be accessible by the JI Integration service. The following table describes the bit values for a IBM 3270 field attribute byte.

Bit Position(s)	Description
6 - 7	Reserved
5	0 = unprotected field 1 = protected field
4	0 = all data allowed (alphanumeric field) 1 = 0 - 9, comma, period, minus and Dup key allowed (numeric field)
2 - 3	00 = normal intensity, not light pen detectable 01 = normal intensity, light pen detectable 10 = high intensity, light pen detectable 11 = non-displayable, not light pen detectable
1	Reserved
0	0 = field has not been modified 1 = field has been modified

## IBM 5250 Field Attribute Byte

---

When an JI Integration service sets the ATTRIBUTES Protocol Agent option to ON for the TN5250 Protocol Agent, IBM 5250 field attribute characters in the presentation space will be accessible by the JI Integration service. The following table describes the bit values for a IBM 5250 field attribute byte.

Bit Position(s)	Description
7	0 = Presentation space position specified a field attribute byte.  1 = Presentation space position did not specify a field attribute byte.
6	0 = non-displayable field  1 = displayable field
5	0 = unprotected field  1 = protected field
4	0 = normal intensity  1 = high intensity
1 - 3	000 = all data allowed (alphabetic shift field)  001 = A - B, a - b, comma, period, minus, blank and Dup key allowed (alphabetic field only)  010 = all data allowed (numeric field only)  011 = 0 - 9, comma, period, plus, minus, blank and Dup key allowed (numeric field only)  100 = Reserved  101 = 0 - 9 and Dup key allowed (digits only field)  110 = only magnetic stripe reader data allowed (signed numeric field)  111 = 0 - 9, plus, minus and Dup key allowed (signed

Bit Position(s)	Description
0	0 = field has not been modified 1 = field has been modified

## IBM 3270/5250 Interpreted Field Attribute Byte

---

When an JI Integration service sets the ATTRIBUTES Protocol Agent option to ON\_INTERPRETED for the TN3270 and TN5250 Protocol Agents, then “interpreted” IBM 3270/5250 field attribute characters in the presentation space will be accessible by the JI Integration service. This is useful when the service needs to access field attribute information but the host application displays screen data using a national language character set other than US-ASCII. The interpreted field attribute information will occupy the range from 0x90 to 0x9f, thereby NOT conflicting with any displayable characters in a national language character set such as those described by the ISO-8859 standards. The following table describes the bit values for the interpreted field attribute byte.

Bit Position(s)	Description
7	1
6	0
5	0
4	1
3	0 = unprotected field 1 = protected field
2	0 = all data allowed (alphanumeric field) 1 = 0 - 9, comma, period, minus and Dup key allowed (numeric field)
1	0 = normal intensity 1 = high intensity

Bit Position(s)	Description
0	0 = displayable field
	1 = non-displayable field

**Note:** The JI Integration include file `EA_keys.h` which resides in the directory `<EA_HOME>/include` (where `<EA_HOME>` represents the value of the `EA_HOME` parameter in the `environment.ccf` configuration file) contains `#define` definitions for the bit mask values which are described in this appendix.



## Glossary

---

<b>ACL</b>	Access Control List
<b>Actions</b>	Used by JI Integration Java services to navigate from one legacy screen to the next on the legacy application. An action includes all data that was input by the user during trail recording in MapMaker, along with the AID key or Action that caused the screen transition.
<b>Action Key</b>	A key sequence that performs an action in the legacy application. Action keys are valid for the Telnet protocol and are similar in concept to AID keys.
<b>Applet</b>	A program written in the Java programming language that is accessed from a Web browser.
<b>Application</b>	A Java program run as a stand-alone program.
<b>API</b>	API - Application Programming Interface. The library of C or Java functions callable from UNIX and Windows programs. Used to develop JI Integration clients and services.
<b>AID Key</b>	The Attention Identifier Key (AID). A single key on the keyboard that, when pressed by the user, performs an action in the legacy application. Typical AID keys include the Enter and PF keys, although the legacy application may change their usage or use other AID keys. AID keys are valid for the TN3270 and TN5250 protocols.
<b>Browser</b>	A program that allows users to access information on a Web server. Also known as a Web browser.
<b>CGI</b>	Common Gateway Interface. A standard method for external gateway programs to interface with Web servers.

<b>Character Encoding</b>	The format or encoding of a language-set character. Character encodings are usually 1, 2, 3, or 4 bytes. Unicode is an example of a 2-byte character encoding. Other examples are ASCII, EBCDIC, and UTF-8.
<b>Character Mode</b>	Character mode describes the functionality in JI Integration that communicates with character-based applications over the Telnet protocol.
<b>Client</b>	<p>In JI Integration, client refers to one of two items:</p> <ul style="list-style-type: none"><li>• A Runtime version of an application developed in Java that uses JI Integration client APIs to communicate with JI Integration services.</li><li>• Also refers to a third-party software application that interfaces with JI Integration services and functions similarly to an application developed with a JI Integration client API.</li></ul>
<b>Client Functions</b>	The C or Java functions used to allow clients to connect to services, execute service methods, and input and extract data.
<b>Content Pane</b>	A content pane, also called a panel, is a GUI component that acts as a container for various GUI components. A content pane is basically a window that other GUI objects, such as buttons and text fields, are placed on.
<b>Cookie</b>	A general mechanism used by Web servers to both store and retrieve information on the client side of the connection.
<b>Custom Classes</b>	Classes that are used to “extend” or customize service code that was generated in MapMaker.
<b>Data Field</b>	A data field is an individual field on the legacy screen that is added to either a data template or a table template. Data fields are used in conjunction with output variables to extract data from the legacy screen.
<b>Data Mapping</b>	Refers to the mapping of data in the flow of a method. Every point in a method where data is sent to or retrieved from an external source requires data mapping. Data mapping is defined in the Data Mapping Editor.

<b>Data Stream</b>	The flow, or stream, of information between computer programs. Data on the data stream is represented using “character encoding” and is transferred using a mutually agreed upon protocol.
<b>Data Template</b>	A data template is a logical representation of non-repeating data fields on the legacy screen. Data templates are defined in MapMaker and are used in conjunction with output variables to extract data from the legacy application. Similar to table template, used to define repeating data fields.
<b>Data Typing</b>	Refers to the creation and definition of data types. Data types are defined and maintained in MapMaker’s Business Entity Editor.
<b>DBCS</b>	Double-Byte Character Set.
<b>DLL</b>	Dynamically Linked Library. A library of function calls used in Windows environments.
<b>DOM</b>	Document Object Model (aka “random access” protocol for XML) – an XML parser that converts the XML document into a collection of objects, which can then be manipulated in any way you choose.
<b>DTD</b>	Document Type Definition for XML – an optional part of the XML document prolog that specifies the kinds of tags that can be included in an XML document and the valid arrangement of those tags.
<b>EAServiceBean</b>	The interface between JI Integration Java service code and the JService that manages the service in the JI Integration server environment. The EAServiceBean can be extended or customized to change the interface if required.
<b>ECS</b>	Extended Character Support.

<b>EIS</b>	Enterprise Information System - an application providing information of critical importance to the day-to-day planning and/or operation of a business. EISs are generally run on larger platforms, such as mainframes or minicomputers. EISs provide the information infrastructure for an enterprise. Examples of EISs include enterprise resource planning systems, mainframe transaction processing systems, relational database management systems, and other legacy information systems.
<b>Enterprise System</b>	A system involved in an organization's critical business processes. Typically, enterprise systems are large and complex, use database management systems (DBMSs), and run on mainframes or minicomputers.
<b>Formatted Fields</b>	Fields on the legacy application that have special formatting characteristics. MapMaker identifies all such fields on the legacy screen and uses the field layout to match screens (unless the fields are disabled and Tags are used to identify screens).
<b>GBBasic</b>	A Java package, <i>com.jacada.mapstudio.GBBasic</i> , that is included with JI Integration and can be used to extend or customize Java service code that was generated in MapMaker.
<b>GUI</b>	Graphical User Interface. An application that allows users to interface with computer programs in a graphical environment. In JI Integration, MapMaker, the Configuration Manager, and the System Monitor are all Graphical User Interfaces.
<b>HTML</b>	HyperText Markup Language, a format used to create Web documents.
<b>Hos</b>	A computer machine where applications reside. In JI Integration, hosts can be the machine on which components of the JI Integration environment are running, the machine on which the telnet, TN3270, or TN5250 server reside, or the machine on which the legacy applications reside.
<b>IBE</b>	Internal Business Entity. Refers to data types that are used internally by the JI Integration Service. A global variable may be defined of an IBE data type.
<b>IDE</b>	Integrated Development Environment. Refers to a graphical development tool that uses standard GUI components to facilitate application development.

<b>Jacada Integrator</b>	See JI Integration.
<b>Java</b>	An object-oriented, platform-independent programming language developed by Oracle.
<b>Java services</b>	JI Integration services developed using the Java programming language. Java services are generated from the MapMaker graphical development interface (GUI) and can be customized using the custom service classes included with JI Integration.
<b>JClient3</b>	The Java Client Library version 3 is an improved version of the JClient, which allows JI Integration clients to be developed using JDK 1.4.2_05 or newer.
<b>JDBC</b>	Java DataBase Connectivity.
<b>JDK</b>	Java Development Kit. The development environment for the Java programming language. Includes a Java Runtime Environment.
<b>JRE</b>	Java Runtime Environment. The minimum environment required to run Java applications. This is a combination of a JVM along with the core classes and files required to run Java applications.
<b>JVM</b>	Java Virtual Machine. A Java interpreter that converts Java code into executable code.
<b>Legacy data</b>	Data residing on a mainframe platform.
<b>Legacy host</b>	The machine or host on which legacy applications reside.
<b>Map</b>	The logical representation of the screens, fields, data input and AID keys that make up the user interaction with a legacy application. Maps are created in MapMaker. Note that JI Integration's use of the term Map is distinct from the java.util.Map that is included with Java.

<b>MapMaker</b>	The graphical development interface provided with JI Integration for the purpose of developing Java services. Used to record trails, maps, data and table templates, create methods and services, and then generate and optionally deploy the services into the JI Integration server environment.
<b>Methods</b>	Object-oriented entity of one or more functions. A collection of methods, initialization code, and events make up a service.
<b>MLM</b>	Map-List-Map. Refers to an XBE data type used for communication with a client, such as a Java, C, or VB Client.
<b>Multicasting</b>	A connectionless IP networking communication in which applications on the IP network broadcast information over a well-known socket.
<b>Multithread-safe</b>	See thread-safe.
<b>Multithreaded</b>	See threaded.
<b>Offset</b>	Refers to the location of data on the legacy screen. The offset is determined using the following format: For an 80 column screen, the offset is (column # - 1) + 80 x (row # - 1). For example, the first column of the second row is position 80: (1 - 1) + 80 x (2-1).
<b>Package</b>	A collection of java classes that are grouped together to form a logical combination of classes.
<b>Presentation Space</b>	A representation of the communications between the legacy host to the JI Integration environment, including the screen and field information from the legacy application.
<b>Protocol Agent</b>	A software interface that governs the procedures used to exchange information between physically remote entities such as computer systems. The Protocol Agent governs the format of the messages, the generation of checking information, and the flow control, as well as the actions to take in the event of errors.
<b>Proxy Server</b>	Special instances of Resource Servers that allow multicasting communication to take place over multiple sub-nets.

<b>Resources</b>	The components of JI Integration that are managed by the Resource Server. These components include Resource Databases and license files.
<b>Resource Database</b>	A database that is used in the JI Integration environment to store environment and service configuration, along with service code and maps.
<b>Resource Serve</b>	rManages the communication between environment managers and the JI Integration resources.
<b>RMI</b>	Remote Method Invocation. A standard from Oracle that allows distributed Java objects to communicate with each other over TCP/IP networks.
<b>Screen</b>	The screen, as contained in the data stream, that is coming from the legacy host.
<b>Screen Mapping</b>	The process of marking the physical boundaries of, and defining the screen components found on, an external application screen. The components include tags, fields, areas, repeating areas, and repeating fields.
<b>Service</b>	A collection of methods which answer requests from a client.
<b>SOCKS</b>	SOCKS is a firewall proxy protocol.
<b>System Monitor</b>	The tool used to monitor the JI Integration System. It allows you to monitor, log, and view real-time activity for all or selected Environment Managers, JClusters and JServices, clients, and services.
<b>Table Template</b>	A Table Template is a logical representation of the area on a legacy screen that contains repeating Data Fields. Table templates are defined in MapMaker and are used in conjunction with output variables to extract data from the legacy application. Similar to Data Template, used to define non-repeating Data Fields.
<b>Tag</b>	A user-defined component of the host application screen that most often serves as a label for fields. You can define any static screen text as a tag. MapMaker can identify external application screens by the tags that are defined for them.

<b>Telnet</b>	A TCP/IP-based terminal emulation protocol. Requires a Telnet server. Telnet is also used to describe the Character Mode functionality within JI Integration.
<b>Terminfo</b>	UNIX terminal information database. See the UNIX terminfo(4) man page.
<b>Thread-safe</b>	Also multithread-safe (MT-safe). A description of a function or library that may be called in a threaded environment without any additional coding.
<b>Thread</b>	A single flow of control within a process or address space. Programs using two or more threads are referred to as threaded or multi-threaded.
<b>Threaded</b>	Also multithreaded. A form of multi-tasking that uses multiple independent execution threads.
<b>TN3270</b>	An implementation of the telnet protocol that is used to communicate between TCP/IP networks and IBM mainframe applications that use IBM 3270 terminals. A TN3270 server is required for connection from the TCP/IP network to the mainframe.
<b>TN5250</b>	An implementation of the telnet protocol that is used to communicate between TCP/IP networks and IBM AS400 applications that use IBM 5250 terminals. A TN5250 server is required for connection from the TCP/IP network to the AS400.
<b>Trail</b>	The linear path of all screens encountered during navigation through a host application. MapMaker records trails during host application interaction.
<b>Unicode</b>	A universal character code.
<b>Web</b>	A network of computers based on the client-server model. The Web uses Web browsers to access information from a Web server. A Web can be a part of the World Wide Web or can be a part of a separate network, also known as an "Intranet".
<b>Web browser</b>	A program that allows users to access information on a Web server.



<b>JI Integration</b>	Consists of one or more Environment Managers, Resource Servers, resources including the Resource Database, and JI Integration clients and services.
<b>XBE</b>	eXternal Business Entity. Refers to data types that are used for the JI Integration Service to send and receive data to and from an external source. Such external sources are Legacy Screens and Clients.
<b>XML</b>	eXtensible Markup Language – a text-based markup language that is fast becoming the standard for data interchange on the web.
<b>XSD</b>	XML Schema Definition. Specifies how to formally describe the elements in an XML document. One of the XBE data types supported by MapMaker is XML/XSD.



---

# Index

## A

Action Key .....	199
Actions .....	199
AID Key .....	199
API .....	199
Applet .....	199
Application Programming Interface .....	199

## B

Browser .....	199
---------------	-----

## C

CGI .....	199, 200
Character Encoding .....	200
Character Mode .....	200
character sets .....	165
Client .....	200
functions .....	200
Configuration files .....	131
Configuration Manager	
adding and editing a license key .....	182
Configuration file .....	145
Cookie .....	200

## D

Data Field .....	200
Data Mapping .....	200
Data Stream .....	201
Data Templates .....	201
Data Typing .....	201
DLL .....	201
Document Object Model (DOM) .....	201
Document Type Definition (DTD) .....	201

Documentation .....	6
Viewing on-line .....	8
DOM .....	201
DTD .....	201

## E

ea_admin .....	10
ea_cfgmgr .....	14
ea_createdb .....	20
EA_ENV .....	148
ea_envmgr .....	29
ea_exportdb .....	32
ea_importdb .....	35
EA_keys.h .....	198
ea_mapmaker .....	38
ea_mapplayer .....	55
ea_ping .....	57
ea_ressvr .....	60
ea_shutdown .....	62
ea_start .....	68
ea_status .....	72
ea_sysmon .....	75
EAServiceBean .....	201
ECS .....	201
Environment Manager	
Configuration file .....	134
Environment Settings .....	148
Environment variables .....	148
environment.ccf configuration file .....	149
Error Messages	
JI Integration .....	87
Error messages .....	87
Extensible Markup Language (XML) .....	207

---

## F

Fields	
formatted	202
Formatted Fields	202
Formatting Conventions	6

## G

GBBasic	202
GUI	202

## H

Host	202
HTML	202

## I

IBE	202
IBM 3270	195
IBM 5250	196
IDE	202
Installation Program	
Uninstalling	193

## J

Java	
runtime environment (JRE)	203
virtual machine (JVM)	203
Java Development Kit (JDK)	203
Java Runtime Environment (JRE)	203
Java Services	203
Java Virtual Machine (JVM)	203
JClient	203
JDBC	203
JDK	203
JI Integration	
error messages	87

messages	87
Uninstalling	193
JI Integration environment	203, 207
JRE	203
JVM	203

## L

Legacy Data	203
Legacy Host	203
License	
adding and editing a license key	182
checking statuses	185
compatibility	191
config.fromUser	189
config.licenseCheckTime	189
config.mailHost	189
config.mailUser	189
count	180
duplicates	190
ea_admin jclusters -e	186
ea_admin licenseStatus	185
ea_admin_status	185
ea_start	185
envmgr.cfg	181
expiration email notification	187
expire flag	180
feature type	179
JI_CLIENT	179
JI_DEMO	179
JI_HOST	179
JI_MAPMAKER	179
JI_SERVER	179
JI_SERVICE	179
key	178
license.txt	181
NO_LICENSE	179
priority	180

---

## M

Map .....	203
MapMaker .....	204
Methods .....	204
Multicast .....	204
Multithreaded .....	204
Multithread-safe .....	204

## O

Offset .....	204
--------------	-----

## P

Package .....	204
Presentation Space .....	204
Protocol Agent .....	204
Protocol agent options .....	195
Proxy Server .....	204

## R

Resource Database .....	205
Resource Manager .....	205
Resource Server Configuration file .....	139
Resources .....	205
RMI .....	205

## S

Screen .....	205
mapping .....	205
Service .....	205
SOCKS .....	205
System Monitor .....	205
Configuration file .....	147

## T

Table Template .....	205
Tag .....	205
Telnet .....	206
Terminfo .....	206
Thread .....	206
Threaded .....	206
Thread-safe .....	206
TN3270 .....	195, 206
TN5250 .....	206
Trail .....	206
Troubleshooting messages .....	87

## U

Unicode .....	206
Uninstalling .....	193

## W

Web .....	206
Web Browser .....	206

## X

XBE .....	207
XML .....	207



---







