

Integrity

MKS Integrity
2006

*Integrations Builder
Guide*

MKS Integrity 2006 Integrations Builder Guide

Copyright © 2001–2006 MKS Software Inc.; in Canada copyright owned by MKS Inc. All rights reserved.

MKS makes no warranty of any kind with regard to this material, including, but not limited to the implied warranties of merchant ability, performance, or fitness for a particular purpose. MKS shall not be liable for errors contained herein, or for any direct, indirect, incidental, or consequential damages resulting from the use of this material.

No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form by any means, without written permission from MKS.

MKS, MKS Source Integrity, MKS Integrity Manager, Implementer, MKS Toolkit, Sandbox, N^UTCRACKER, MKS Integrity Solution, AlertCentre, MKS Integrity Suite, and MKS Federated Server are trademarks or registered trademarks of MKS Inc. All other trademarks or registered trademarks are the property of their respective holders.

Corporate Headquarters

410 Albert Street
Waterloo, ON N2L 3V3
Canada
tel: 519 884 2251
fax: 519 884 8861
sales (toll free): 800 265 2797
www.mks.com

Worldwide Offices:

1815 South Meyers Rd.
Suite 220
Oakbrook Terrace, IL USA
60181
tel: 630 827 4900
fax: 630 629 9167
sales (toll free): 800 633 1235
12450 Fair Lakes Circle
Suite 400
Fairfax, VA USA
22033
tel: 519 884 2251
fax: 703 803 3344
sales (toll free): 800 637 8034

Martinstraße 42-44
73728 Esslingen
Germany
tel: +49 711 351775 0
fax: +49 711 351775 7555
Third Floor, Duke's Court
Duke Street, Woking
Surrey
GU21 5BH
United Kingdom
tel: +44 (0)1483 733900
fax: +44 (0)1483 733901
sales: +44 (0)1483 733919

Table of Contents

Chapters

1	Overview	1
	About This Guide	2
	Related MKS Integrity Documentation	2
	Typographical Conventions	5
	Assumptions	5
	Getting Help	6
	Professional Services	7
	Your Feedback Is Welcome	8
2	Understanding the MKS API	9
	The API Model	10
	Integration Scenarios	11
	Integrating With the Integrity Client	12
	Integrating With the Integrity Server	13
	Integrating With the Integrity Client as Server	13
	Integrating Your Desktop With the Integrity Server	14
	Where To Go Next	15
3	Using the MKS API	17
	API Documentation	18
	Java API Documentation	18
	C API Documentation	18
	Setting Up Your Environment	19
	Java API Environment	19
	C API Environment	19
	Integration Objects	20
	Integration Points	20
	Controlling Access to an Integration Point	20
	Communicating With an Integration Point	20
	Creating an Integration Point	21
	Sessions	22
	Session Caching	22
	Using a Common Session	23
	Creating a Session	23
	Command Runners	23
	Running Commands	24
	Specifying a Command	24
	Default Command Behavior	25
	Executing a Command	26

Reading Responses	26
Creation Commands	26
Action Commands	27
View Commands	28
Command Response Summary	31
Viewing Responses With the API Viewer Utility	32
Java API Exceptions	34
Catching Exceptions	35
Work Item Exceptions	36
API Connection Exceptions	36
Command Exceptions	36
Application Exceptions	37
Internal Exceptions	39
Integrity Client Launch Exceptions	39
Interrupted Exceptions	39
C API Errors	39
C API Application Errors	39
Internal Error Codes	41
Command Variations	42
Using the Graphical User Interface	43
GUI for Status Reporting	43
GUI for Prompting	43
Full GUI	44
Using Interim Responses	44
Using Impersonation	45
Using File References	46
Integration Point	46
Current Working Directory	46
Location of the API	47
Using Uncertified Commands	47
Best Practices	47
Copy the API	48
Managing Users	48
Single User	48
Multiple Users	50
Dedicated Sessions	52
Memory Management Tips	53
Use Facade Patterns	54
Converting MKS Source Scripts to API Programs	54
Accessing Item Fields	55
Retrieving Item Fields as Strings for the Java API	55
Retrieving Item Fields as Strings for the C API	55
Creating Programs for the C API	56

4	API Certified Commands	59
	Format for Command Information	60
	Certified Commands	61
	MKS Integrity	61
	MKS Source	61
	Standard Data Types	62
	Unicode Support for C API	63
	User Item Data Type	63
	Date and Datetime Data Types	63
	Integrity Manager Commands	64
	about	64
	columnsets/viewcolumnset	64
	connect	64
	copyissue	64
	cpattributes/viewcpattribute	64
	cpcentryattributes/viewcpcentryattribute	65
	cps/viewcp	67
	cptypes/viewcptype	67
	createcp	68
	createcpattribute	68
	createcpcentry	69
	createcpcentryattribute	69
	createcptype	69
	createissue	70
	deletcp	70
	deletcpattribute	70
	deletcpcentry	70
	deletcpcentryattribute	71
	deletcptype	71
	disconnect	71
	editcpattribute	71
	editcpcentry	72
	editcpcentryattribute	72
	editcptype	73
	editissue	73
	exit	73
	extractattachments	73
	fields/viewfield	74
	groups/viewgroup	74
	importissue	74
	issues/viewissue	74
	projects/viewproject	75
	queries/viewquery	76
	servers	76
	states/viewstate	77
	types/viewtype	77
	users/viewuser	77
	viewcolumnset	78
	viewcp	78

viewcpattribute	78
viewcpentryattribute	78
viewcptype	78
viewfield	78
viewgroup	79
viewissue	79
viewproject	79
viewquery	79
viewstate	79
viewtype	79
viewuser	80
Source Integrity Commands	80
about	80
add	80
addsubproject	81
archiveinfo	81
ci	82
closecp	82
co	83
connect	83
configuresubproject	83
createcp	84
createproject	84
createsandbox	84
createsubproject	84
disconnect	85
drop	85
exit	85
gui	86
lock	86
locks	86
projectadd	86
projectci	87
projectco	87
projectinfo	88
rename	89
resync	89
revert	90
sandboxinfo	91
unlock	92
viewhistory	93
viewproject	93
viewsandbox	94

5 Administrating the MKS API 95

 Setting Up Permissions 96

 Setting Up Policies 96

 Client Integration Point Policy Defaults 96

 Server Integration Point Policy Defaults 97

 Client as Server Integration Point Policies 97

 Setting Up Impersonations 97

 Backward Compatibility 98

 Java and C API Compatibility 98

 Command Compatibility 99

 Patch Management 99

 Troubleshooting 100

 Using the API Viewer 100

 API Version Number 100

 Using API Logging 100

 Supported Configurations 101

Appendixes

Index 103

Product Notices 109

Overview

Understanding and Using This Guide

The *Integrations Builder Guide* gives you the information you need to build your own integrations using the MKS application programming interface (API). It provides an overview of the MKS API, instructions and guidelines to help you start coding your integration, and details of how to set up the MKS Integrity for the API.

Specifically, this chapter discusses the following topics:

- “About This Guide” on page 2
- “Related MKS Integrity Documentation” on page 2
- “Typographical Conventions” on page 5
- “Assumptions” on page 5
- “Getting Help” on page 6
- “Professional Services” on page 7
- “Your Feedback Is Welcome” on page 8

About This Guide

This guide is divided into the following chapters:

- **Chapter 2: “Understanding the MKS API” on page 9**

Provides information to help you understand how the MKS API can meet your integration needs.

Before proceeding with developing your integration, you may find it useful to read this chapter to understand some of the concepts and features of the MKS API.

- **Chapter 3: “Using the MKS API” on page 17**

Provides information to help you start developing your integrations using the Java and C versions of the MKS API.

- **Chapter 4: “API Certified Commands” on page 59**

Provides work item and result information for all certified API commands.

- **Chapter 5: “Administering the MKS API” on page 95**

Provides information on how to set up and administrate the MKS API.

Related MKS Integrity Documentation

MKS Integrity documentation included in this release is as follows:

- *MKS Integrity Server 2006 Installation Guide* provides the information required to build a basic understanding of MKS Integrity, and to install and configure the solution.
- *MKS Integrity Server 2006 Upgrading Guide* describes important aspects of upgrading that you should be aware of, as well as other pre- and post-installation considerations.
- *MKS Integrity Server 2006 Administration Guide* provides information on post-installation tasks: configuring and administering the MKS Integrity Server, and how to configure and set up MKS Integrity and MKS Source.
- *MKS Integrity Server 2006 Administration CLI Reference Guide* provides details on the command line utilities available for administration tasks.
- *MKS Integrity Server QuickStart Guide* describes how to perform a simple installation of MKS Integrity Server using its default settings.
- *MKS Source 2006 User Guide* tells users how to get the most out of MKS Source and explains how to approach day-to-day tasks.
- *MKS Source 2006 CLI Reference Guide* gives details on the command line utilities included with MKS Source.

- *MKS Integrity 2006 User Guide* tells users how to get the most out of MKS Integrity and explains how to approach day-to-day tasks.
- *MKS Integrity 2006 CLI Reference Guide* gives details on the command line utilities included with MKS Integrity.
- *MKS Integrity 2006 Integrations User Guide* describes how to access the functionality of MKS Source and MKS Integrity while working within your favorite integrated third party tool, such as Sybase PowerBuilder and Microsoft Project.
- *MKS Integrity 2006 Integrations Builder Guide* provides the information you need to create your own integrations through the Application Programming Interface (API).
- *MKS Integrity Solution for Requirements User Guide* provides details on how to install and use MKS Requirements.
- *MKS Integrity Solution for ITIL User Guide* provides details on how to install and use ITIL.
- *MKS Integrity Solution for Waterfall Development User Guide* provides details on how to install and use the MKS Integrity Solution designed for a corporate IT department environment.
- *MKS Integrity Solution for Iterative Development User Guide* provides details on how to install and use the MKS Integrity Solution designed for a software development environment.
- *MKS Deploy 2006 Administration Guide* is based on a start-to-finish approach to setting up and maintaining a staging system. It mirrors a logical MKS Deploy (with MKS Agent) configuration, staging system creation, and maintenance process.
- *Learning the MKS Integrity Client Interface* describes how the client interface has changed since MKS Integrity Solution 4.6.
- *Using MKS Make* describes the MKS Make tool offering developers, project managers, and authors an efficient way to automate the production and maintenance of any project, large or small. Whenever you make changes to an element of a development project, MKS Make automatically recompiles all related files and no others, saving valuable time and eliminating errors.
- *Release Notes* provide the most up-to-date information about the release. You should read them before installing and running any MKS Integrity Client component or the MKS Integrity Server. The information also includes fixed issues and known issues.

You can read the release notes from the DVD Browser as HTML documents in a Web browser.

NOTE In addition, you can browse to www.mks.com/products/whitepapers.shtml to view and download white papers that cover best practices and more in-depth applications of MKS Integrity.


To provide you with the most convenient means of retrieving information, product documentation is available in several formats: print, Adobe Acrobat's Portable Document Format (PDF), online help, and manual pages (man). The following table indicates the available formats for each document.

Documentation Title	Print	PDF	Online	Man
<i>MKS Integrity Server 2006 Installation Guide</i>	Yes	Yes	AA,I,S ^a	No
<i>MKS Integrity Server 2006 Upgrading Guide</i>	Yes	Yes	No	No
<i>MKS Integrity Server 2006 Administration Guide</i>	Yes	Yes	AA,I,S ^a	No
<i>MKS Integrity Server 2006 Administration CLI Reference Guide</i>	No	Yes	Yes	Yes
<i>MKS Integrity Server QuickStart Guide</i>	Yes	Yes	No	No
<i>MKS Source 2006 User Guide</i>	No	Yes	Yes	No
<i>MKS Source 2006 CLI Reference Guide</i>	No	Yes	Yes	Yes
<i>MKS Integrity 2006 User Guide</i>	No	Yes	Yes	No
<i>MKS Integrity 2006 CLI Reference Guide</i>	No	Yes	Yes	Yes
<i>MKS Integrity 2006 Integrations User Guide</i>	No	Yes	Yes	No
<i>MKS Integrity 2006 Integrations Builder Guide</i>	No	Yes	No	No
<i>MKS Integrity Solution for ITIL User Guide</i>	No	Yes	No	No
<i>MKS Integrity Solution for Requirements User Guide</i>	No	Yes	No	No
<i>MKS Integrity Solution for Waterfall Development User Guide</i>	No	Yes	No	No
<i>MKS Integrity Solution for Iterative Development User Guide</i>	No	Yes	No	No
<i>MKS Deploy 2006 Administration Guide</i>	No	Yes	No	No
<i>Learning the MKS Integrity Client Interface</i>	No	Yes	Yes	No
<i>Using MKS Make</i>	No	Yes	No	No
<i>Release Notes</i>	No	No	Yes	No

^a For Authorization Administration graphical user interface, and MKS Integrity and MKS Source policies only.

PDF files are located in the `\documentation` subdirectory of the distribution DVD. To view them you must have the Acrobat Reader™ installed on your machine. Acrobat Reader is available at <http://www.adobe.com/products/acrobat/>. Once you install Acrobat Reader, whenever you open a PDF file Acrobat Reader starts automatically.

Online help is accessible from within the graphical user interface and the Web interface. You can access online help in the following ways:

- Select **Contents** from the **Help** menu.
- Click  in the toolbar.
- Press F1 (GUI only).

Online *man* (manual) pages for the command line utilities are available on the client by using the **man** command in the command line interface.

Typographical Conventions

Throughout the documentation, the following typographical conventions identify the features, functions, and components of the MKS API.

Items in documentation...	Appear as...
Commands	si gui
File names and paths	<code>\mksis\lib\mksapi.jar</code>
Classes and structs	<code>CmdRunner</code>
Methods and functions	<code>getNewCommandRunner()</code>
New terms	appear in <i>italics</i> the first time

NOTE A note provides you with information that supplements the key points of the subject. A note may also supply information that applies only in particular cases.

IMPORTANT An important note provides you with information that is essential for completing a task.

CAUTION A caution note advises you about situations that have the potential to result in a loss of data.

Assumptions

Before using the MKS API, MKS assumes the following about your knowledge and experience:

- You understand Java, if you are building an implementation using the Java version of the MKS API.
- You understand C, if you are building an implementation using the C version of the MKS API.
- You understand the applications you are integrating with the MKS Integrity.
- You understand the hardware platforms and operating systems that your applications are installed on and that the MKS Integrity is installed on.
- You are comfortable with the command line interface (CLI) for both MKS Integrity and MKS Source.

Getting Help

MKS Customer Care is focused on delivering the right solutions to issues as they arise. For assistance, you can choose online support or telephone a Technical Support Representative.

For online support, browse to www.mks.com/support. There you will find easy access to e-mail, Web request services, automatic product notifications, and the MKS Customer Community – a secure database that provides helpful resources such as product documentation, knowledge base articles, product downloads, user forums, presentations, and more.

MKS’s global support professionals comprise a tightly knit team of problem solvers, sharing critical information to help you resolve issues in the shortest possible time with optimal results. Support representatives can provide you with a variety of product related tips and innovative solutions to your unique requirements.

Online		Web	www.mks.com
		E-mail	support@mks.com
Telephone	Toll Free	North America	800 219 4842
		Outside North America	800 219 48424
		India	000 800 100 6273
	Direct	North America	519 884 2270
		UK	+44 (0) 1483 733910
		Germany	+49 711 351775 7575
Fax	North America		519 884 8861
	UK		+44 (0) 1483 733901
	Germany		+49 711 351775 7555

The hours of operation for Customer Care in your region are as follows:

- North America: 8:00 A.M. to 8:00 P.M., Monday to Friday, Eastern Standard Time (GMT-5)
- United Kingdom: 9:00 A.M. to 5:30 P.M., Monday to Friday, British Standard Time (GMT)
- Germany: 9:00 A.M. to 5:30 P.M., Monday to Friday, West Europe Standard Time (GMT+1)
- India: 8:30 A.M. to 5:30 P.M., Monday to Friday, India Standard Time (GMT+5.5)

Professional Services

At MKS, our professional services team is committed to understanding the ever changing development environments of our clients. The professional services team can truly add value to your MKS technology investment by providing the following services that meet your unique requirements:

- process analysis
- software installation
- best practice consultation
- source archive migration
- upgrading
- implementation
- integration between MKS Integrity and other products
- customization of MKS Integrity reports, charts, and dashboards
- product and process training

This ensures that the implementation of your new software goes smoothly, while allowing your developers to make progress on critical projects.

For more information on MKS professional services:

Online	Web	Global	www.mks.com/services
	E-mail	North America	consulting@mks.com
		Europe	europe@mks.com
Telephone		North America	800 633 1235
		UK	+44 (0) 1483 733904
		Germany	+49 711 351775 0

Your Feedback Is Welcome

MKS welcomes your feedback on the documentation included with this product. If you have comments or suggestions about any of the guides or the online help, send them to:

pubs-mgr@mks.com

Include the following information with your feedback:

- product name and version number (from the **About** box)
- title of manual or online help topic
- page number (for manuals only)
- your suggested correction or improvement

NOTE The e-mail address is provided for comments on the MKS documentation only. For technical questions or for software support, you should contact MKS Customer Care (support@mks.com).

Understanding the MKS API

How the MKS API Can Meet Your Integration Needs

MKS recognizes the value of extending the functionality of its two core products – MKS Source and MKS Integrity – by providing a programmatic means of integrating with other tools and applications. The value of the MKS products is in their ability to interact with your source and processes at their point of origin. This often involves integrating with a third party tool, or an internally developed tool or solution, and can span multiple platforms and implementation languages.

To develop integrations and solutions for your specific tool or process, you need the appropriate mechanism. MKS provides multiple integration mechanisms.

- **MKS application programming interface (API)**

This integration mechanism provides Java and C access for running MKS Source and MKS Integrity commands and working with their responses. Complete details are in this manual.

- **MKS API COBOL adapter**

This integration mechanism provides COBOL access to the MKS API from a zSeries. Detailed information is available on the MKS Developer Zone page of the MKS Web site (www.mks.com/devzone).

- **Command Line Interface (CLI) and scripts**

A complete set of commands is available for both MKS Source and MKS Integrity. These allow a user, script, or program with access to a command processor to interact with the MKS products. Scripts are often easy to write for administrators and familiar to users. However, if a script requires parsing the command output, consider using the MKS API, which requires no parsing.

- **Event triggers**

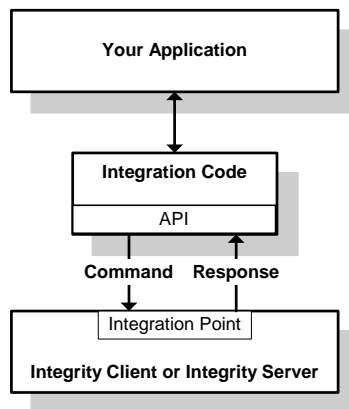
Both MKS Source and MKS Integrity support event triggers. Actions within either product can trigger any function that has been defined as a trigger. See the *MKS Integrity Server 2006 Administration Guide* for details.

This guide is focused on the MKS Java and C API, and its usage. This chapter contains the following information to help you understand how the MKS API can meet your integration needs:

- “The API Model” on page 10
- “Integration Scenarios” on page 11
- “Where To Go Next” on page 15

The API Model

The MKS API uses a generic model to minimize the development and administration of all integrations built using the API. All integrations interact with Integrity products through a generic command processor called the *command runner*. To run any command, a command is specified using its CLI syntax. A command runner sends the command to an Integrity Client or Integrity Server, which executes the command and returns a response. The response contents can then be accessed by the integration.



The same command runner is used to run any command for any application, only the command and its options change, and all commands use the same response type. Rather than learning multiple APIs, there is only a single API to become familiar with.

The MKS API consists of three components:

- **Commands**

The MKS Integrity CLIs define the functions that are available through the API. Key MKS Integrity commands (issue and change package creation and manipulation) and core MKS Source functions, used by most IDE (Integrated Development Environment) integrations, are certified with full API support. Limited default behavior is available for all commands. For more information, see “API Certified Commands” on page 59.

■ Integration Points

The *integration point* is the point in the MKS product where integrations can connect and run commands. You connect and interact with MKS products by running commands on the Integrity Client or the Integrity Server. The same commands are used for both integration points, although the behavior and valid options may change. Commands are run within a session on the integration point.

For more information on sessions, see “Sessions” on page 22. For more information on integration points, see “Integration Scenarios” on page 11.

■ Access Methods

The *access method* is the technology you use to run the API. The access method does not contain any application logic. Currently, you can use Java, C, or COBOL as access methods. For more information on using the Java or C API, see “Using the MKS API” on page 17. For information on how to build integrations using COBOL, see the MKS Developers Zone page on the MKS Web site (www.mks.com/devzone). This Web page also provides information on any additional current or future API access methods.

A specific integration is characterized by how it uses the API model: what commands it runs, on what integration points, using which access method. For example, a typical IDE integration would run MKS Source commands (commands) on the Integrity Client (integration point) using Java (access method); or a help desk integration might run MKS Integrity commands (commands) on the Integrity Server (integration point) using C (access method).

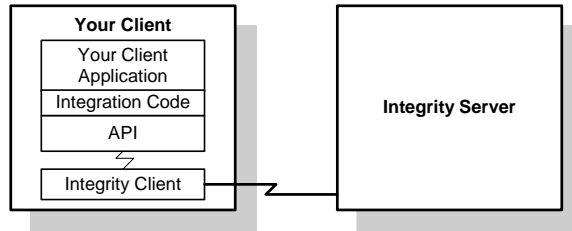
Integration Scenarios

Although it may be easy to determine the commands you need to run and the access method you need to use for your integration, deciding which integration point to use is more involved. The Integrity Client and Integrity Server integration point behave differently depending on the location of the integration relative to the integration point. The most common scenarios are described in this section:

- “Integrating With the Integrity Client” on page 12
- “Integrating With the Integrity Server” on page 13
- “Integrating With the Integrity Client as Server” on page 13
- “Integrating Your Desktop With the Integrity Server” on page 14

Integrating With the Integrity Client

An Integrity Client integration is the most common integration; for example, it is used by all IDE and developer productivity tool integrations.



The Integrity Client integration has many unique features.

- **GUI interaction for status, dialogs, and views**

Integrations can use the Integrity Client GUI when integrating with an Integrity Client. For more information, see “Using the Graphical User Interface” on page 43.

- **Common session**

By using a common session, an Integrity Client integration can provide the user with seamless interaction between the integration, Integrity Client GUI, and the CLI. Changes made through the GUI or CLI can influence the integration and vice versa. For example, establishing a connection using the Integrity Client provides the default connection for the integration.

You also have the option of not using the tight coupling provided by a common session, and using the GUI and CLI independently from the integration.

- **User preferences**

All user preferences defined on an Integrity Client are available to an Integrity Client integration. This is an especially valuable feature because it allows an Integrity Client integration to not have to specify connection or user information, but to rely on the defaults set on the Integrity Client.

- **Integrity Client port detection**

The port of the Integrity Client integration point can be automatically detected when integrating to an Integrity Client installed on the same system as the integration.

- **Local file system access**

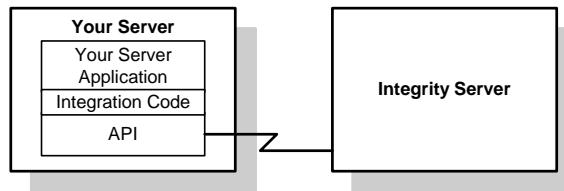
Commands run through the API on an Integrity Client have complete file system access, the same as if they were run directly on the Integrity Client. For more information, see “Using File References” on page 46.

- **Automatically starting the Integrity Client**

The Integrity Client may be automatically started if it is not already running when integrating to an Integrity Client installed on the same system as the integration.

Integrating With the Integrity Server

When no unique Integrity Client functions are required, you can integrate directly with the Integrity Server. Integrating with the Integrity Server eliminates the need for an Integrity Client installation and the administration that is associated with it. This scenario works well when integrating two server-based applications, for example, integrating a help desk with MKS Integrity.

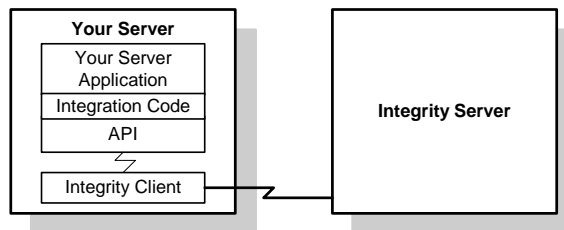


All commands are available but there are some restrictions on Integrity Client-oriented commands and options. For example, directing any output to a GUI causes an error. In addition, commands that contain a file reference satisfy that reference using the Integrity Server file system. For example, checking out a file checks it out on the Integrity Server. For more information, see “Using File References” on page 46.

NOTE Integration with an Integrity Server that is setup as a proxy server is not supported.

Integrating With the Integrity Client as Server

Some server to server integrations require functions that are only available with the Integrity Client. If this is the case, you can install the Integrity Client on your application server. The Integrity Client is capable of running simultaneous commands for multiple users.



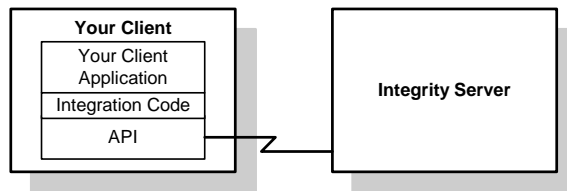
There are two main reasons for using this scenario:

- you need full local file system access (typically used for MKS Source integrations)
- there is an intervening Integrity proxy between the servers

You will need to change your connection and authentication policies to accommodate this scenario. For more information, see “Setting Up Policies” on page 96.

Integrating Your Desktop With the Integrity Server

If your desktop application only needs information from the Integrity Server and nothing unique to the Integrity Client, then you can integrate your desktop directly to the Integrity Server.



This scenario requires the minimum configuration and administration.

Where To Go Next

The following table summarizes the steps you should follow to develop an integration using the MKS API.

To Learn About This...	See...
Guidelines and best practices for developing your integration	"Using the MKS API" on page 17
Commands that you can use in your integration	"API Certified Commands" on page 59
Setting up MKS Integrity for your integration	"Adminstrating the MKS API" on page 95
Information about public classes for Java API	MKS Java API documentation accessed through the MKS Integrity Server Web page
Information about public header files for C API	MKS ANSI C API documentation accessed through the MKS Integrity Server Web page
Sample integration applications, integration issues, and latest information on MKS integrations	MKS Developer Zone page of the MKS Web site (www.mks.com/devzone)
Response and behavior of commands when run through the Java API	<code>com.mks.api.util.APIViewer</code> in the MKS Java API documentation accessed through the MKS Integrity Server Web page
Response and behavior of commands when run through the C API	<code>mksAPIViewer</code> , found in the <code><Integrity client install directory>\bin</code> directory

Using the MKS API

Developing Your Integrations

This chapter contains information to help you start developing your integrations using the MKS API. Specific procedures and/or examples for the Java and C versions of the API are provided where appropriate.

The Java API is contained within `mksapi.jar`, which is installed in the `lib` directory on both the client and server (for example, `<install directory>\lib` for the client, `<install directory>\server\lib` for the server). Make sure that `mksapi.jar` is available on your classpath.

The C API header files are installed on the Integrity client in the `<install directory>\lib\include` directory.

This chapter contains the following information:

- “API Documentation” on page 18
- “Setting Up Your Environment” on page 19
- “Integration Objects” on page 20
- “Running Commands” on page 24
- “Command Variations” on page 42
- “Best Practices” on page 47
- “Creating Programs for the C API” on page 56

API Documentation

Details of all Java public classes and all C functions are provided in separate documents.

Java API Documentation

The Java API public classes are documented in the MKS Java API documentation, which can be accessed through the Integrity Server Web page. If this documentation is not available through the web page, contact your administrator.

The MKS Java API documentation contains the following packages:

- `com.mks.api` contains classes and methods used to run API commands.
- `com.mks.api.response` contains classes and methods used to access API responses.
- `com.mks.api.si` contains model type constants for MKS Source.
- `com.mks.api.im` contains model type constants for MKS Integrity.
- `com.mks.api.ic` contains model type constants for the Integrity Client.
- `com.mks.api.util` contains classes and methods used for testing API commands.

C API Documentation

The C API functions are documented in the MKS ANSI C API documentation, which can be accessed through the Integrity Server Web page. If this documentation is not available through the web page, contact your administrator.

The MKS ANSI C API documentation contains the following packages:

- `mksCommand.h` contains functions used to run API commands.
- `mksError.h` contains the error codes that can be returned by API commands.
- `mksLog.h` contains functions used to log API messages.
- `mksResponse.h` contains functions used to access API responses.
- `mksResponseUtil.h` contains functions used for displaying API responses.
- `mksVersion.h` contains the MKS API version.
- `mksapi.h` includes all of the above.

Setting Up Your Environment

The following environment setup is required for the MKS API.

Java API Environment

You can use the Java version of the API with any platform that has a supported version of the Java Runtime Environment (JRE). For details on the supported versions, see “Supported Configurations” on page 101.

NOTE

- MKS recommends using JRE 1. 5.
 - You cannot use the Java API with Integrity products prior to version 4.6.
-

C API Environment

The C version of the API is implemented using ANSI C. You can use the C API with any ANSI C compiler on supported platforms. For details on the supported platforms, see the *MKS Integrity Server 2006 Administration Guide*.

NOTE You cannot use the C API with Integrity products prior to version 2005.

To use the C API libraries on any UNIX platform, you need to set the environment variable `LD_LIBRARY_PATH=$CLIENT_DIR/lib/$sys` (for HP-UX set `SHLIB_PATH`, for AIX set `LIBPATH`).

For North American UNIX servers to view German characters through the C API, the locale needs to be set correctly using `setlocale()`. If the locale is not set correctly, German characters come out as ?. To correctly set the locale, do one of the following:

- In a shell, `#export LC_ALL=en_US.ISO8859-1`
- In a C program, `setlocale(LC_ALL, "");`

The C API uses the OpenSSL package to communicate with the Integrity Server via SSL when using a secure integration point. This package requires the `/dev/[u]random` device. If your platform does not have this device, one of the following packages can be installed and used:

- EGD: The Entropy Gathering Daemon (<http://egd.sourceforge.net>)
- PRNGD: Pseudo Random Number Generator Daemon (http://www.aet.tu-cottbus.de/personen/jaenicke/postfix_tls/prngd.html)

Integration Objects

Before starting to develop your integration, you need to understand the following three objects:

- “Integration Points” on page 20
- “Sessions” on page 22
- “Command Runners” on page 23

Integration Points

The *integration point* represents the Integrity Client or Integrity Server you want to connect to. The API makes requests of an integration point, and the integration point determines the response to the requests.

Controlling Access to an Integration Point

You control access to an integration point through connection and authentication policies.

The connection policy determines where an integration point accepts a connection from. By default, the client only accepts connections from `localhost` and the server only accepts connections from a specified set of IP addresses.

The authentication policy determines whether you need to specify a user and a password for a session. By default, the client integration point allows connections without specifying a user (as long as connections are limited to `localhost`). The server default requires a user and password.

NOTE You should not confuse the user and password required for a session with the user information specified on the command runner or through the `--user` and `--password` command options.

For more information on connection and authentication policies, see “Setting Up Policies” on page 96.

Communicating With an Integration Point

The API and the integration point communicate using TCP connections. The server uses the standard configured Integrity port to listen for communications requests (7001 by default). The client also uses a configured port, which can be automatically detected by the API.

If you are integrating with the server, you can use a secure port for communications with the API. For information on creating an integration point that uses a secure port, see “Creating an Integration Point” on page 21. For information on how to configure a secure port, see the *MKS Integrity Server 2006 Administration Guide*.

Before you can use an integration point you must create a session. Once you create a session, you receive any exceptions related to the integration point, and its connection and authentication policies.

Creating an Integration Point

Only one integration point needs to be created for each physically different client or server you want to integrate with.

To create an integration point for the Java API

- 1 Create an instance of the `IntegrationPointFactory` class by calling `IntegrationPointFactory.getInstance()`.
- 2 Create an integration point using one of the following methods, depending on whether you are connecting to a local client or a remote client or server:
 - If you are connecting to a local client, create an `IntegrationPoint` by calling the `createLocalIntegrationPoint()` method from the `IntegrationPointFactory` instance.

You can also have the API start the Integrity Client, if it is not already running. For more information, see the MKS Java API documentation.
 - If you are connecting to a remote server or client, create an `IntegrationPoint` by calling the `createIntegrationPoint(String, int)` method from the `IntegrationPointFactory` instance, where the `string` parameter is the host name and the `int` parameter is the port number of the Integrity Client or Integrity Server you want to connect to.

NOTE If you want to create an integration point using a secure communications protocol, call the `createIntegrationPoint` method with the `secure` parameter set to `true`.

To create an integration point for the C API

- 1 Call the `mksAPIInitialize()` function.
- 2 Create an integration point using one of the following functions, depending on whether you are connecting to a local client or a remote client or server:
 - If you are connecting to a local client, create an `mksIntegrationPoint` by calling the `mksCreateLocalIntegrationPoint()` function.

NOTE If you want to have the API start the Integrity Client start automatically if it is not already running, call the `mksCreateLocalIntegrationPoint` function with the `autostart` parameter set to `TRUE`.

- If you are connecting to a remote server or client, create an `IntegrationPoint` by calling the `mksCreateIntegrationPoint(char *host, unsigned int port)`, where the `host` parameter is the host name and the `unsigned int port` parameter is the port number of the Integrity Client or Integrity Server you want to connect to.

NOTE If you want to create an integration point using a secure communications protocol, call the `mksCreateIntegrationPoint` function with the `secure` parameter set to `TRUE`. For information on setting up your environment for a secure integration point, see “C API Environment” on page 19.

Sessions

Commands are run within a *session* on the integration point. Sessions control the interactions of commands being executed on the same integration point. A session has a physical TCP connection between the API and the integration point and can run any command on any server for any application, for any user that the integration point allows.

The main factor in determining how to use sessions is the number of users that the integration is serving. For more information, see “Managing Users” on page 48.

Session Caching

A session caches all application information per user. For example, the Integrity Client caches recent Sandbox® information for quick re-retrieval. A session also caches server connection and state information shared by all users of the session.

On the Integrity Client, a session caches all server connections between the Integrity Client and Integrity Server from the time they are connected until either the session is deleted or the individual connection is disconnected. When a command is run, an existing server connection is used if there is one for the specified server, port, and user. The password is not required nor is it checked.

A session maintains the current state of commands. For example, the current values of command preferences are maintained by the session. Once the current state of a preference is changed through the GUI preferences or the **setprefs** command, the new preference is used for every subsequent command run by any user on that session.

Using a Common Session

A special common session is available only on the Integrity Client. The common session is the standard MKS session available to anyone using the Integrity Client. It is shared by the CLI and the GUI, and is available for integrations as well. This provides a consistent experience for users who interact with the Integrity Client through all three methods. The common session is not restricted to local clients.

Creating a Session

You create a session from an integration point.

To create a session for the Java API

Create a `Session` from the `IntegrationPoint` instance by doing one of the following:

- For a standard session on an Integrity Server, call the `createSession()` method, passing the user and password.
- For a standard session on an Integrity Client, call the `createSession()` method.
- For a common session on an Integrity Client, call the `getCommonSession()` method.

To create a session for the C API

Create a `Session` from the `mksIntegrationPoint` variable by doing one of the following:

- For a standard session on an Integrity Server, call the `mksCreateSession(char *user, char *password)` function, passing the user and password.
- For a standard session on an Integrity Client, call the `mksCreateSession()` function, passing `NULL` in place of a user and password.
- For a common session on an Integrity Client, call the `mksGetCommonSession()` function.

Command Runners

The *command runner* manages the sequential execution of commands on a session. If more than one command needs to be executed at the same time on an integration point, you need to create multiple command runners for a session. Since there can be multiple sessions, it is possible to have an integration using multiple command runners, concurrently running commands in multiple sessions, connected to multiple integration points.

You can set connection-based command options for a command runner that are applied to every command. For more information, see “Managing Users” on page 48.

NOTE Even though not all application commands explicitly allow these connection settings, they do not cause an error if set for any command.

To create a command runner

You create a command runner from a session.

- **For the Java API**

Create an instance of the `CmdRunner` class by calling the `createCmdRunner()` method from the `Session` instance.

- **For the C API**

Create an instance of the `mksCmdRunner` by calling the `mksCreateCmdRunner()` function from the `mksSession` variable.

Running Commands

There are three activities related to running commands using a command runner:

- “Specifying a Command” on page 24
- “Executing a Command” on page 26
- “Reading Responses” on page 26

Specifying a Command

A *Command* specifies the application to interact with and the command name, and can have any number of options and selections.

Anything that can be specified on a command line can be specified using a `Command` class or function.

For details on the options and selections for each command, see the *MKS Integrity 2006 CLI Reference Guide* and the *MKS Source 2006 CLI Reference Guide*. For details on the responses returned by each command, see “API Certified Commands” on page 59.

To specify a command for the Java API

You can either specify the command with the `Command` class and related classes or through a `String` array.

For example, for the **si about** command, you could either pass the `String` array:

```
String[] cmd =
    new String[] {"si", "about"};
```

Or you could construct the command using the following:

```
Command cmd = new Command(Command.SI, "about");
```

For example, you could construct the **im issues** command with the following options:

```
Command issuesView = new Command(Command.IM, "issues");
issuesView.addOption(
    new Option("fields", "ID,Type,Summary"));
issuesView.addOption(
    new Option("query", "Quick Query"));
```

NOTE Unlike CLI commands, options for commands used in the API do not require a “--” prefix, if the commands are created using the `Command` class. The prefix is still required if `String` arrays are used.

To specify a command for the C API

You can either specify the command by creating a `char *` array or through an `mksCommand` structure.

For example, for the **im viewissue** command, you could either create a `char *` array:

```
wchar_t* cmd[] = {"im", "viewissue", "--gui"};
```

or you could construct the command using an `mksCommand` structure:

```
mksCommand = cmd;
mksrtn = rtn;

rtn = mksCreateCommand(cmd);
cmd->appName = "im";
cmd->cmdName = "viewissue";

mksOptionListAdd(cmd->optionList, "--gui", NULL);
```

Default Command Behavior

If a specific behaviour is required of a command, MKS recommends specifying all the options needed in the command rather than relying on defaults set through command preferences.

You can use the GUI to specify command settings for any command run through the API by specifying the option `--settingsUI=gui`. For more information, see “Using the Graphical User Interface” on page 43.

Some preferences can only be changed through the CLI **si setprefs** command, using the `--ui` option. For more information, see **si viewprefs** and **si setprefs** in the *MKS Source 2006 CLI Reference Guide*.

There are also API-specific default preferences that are used in special situations. For example, the default set of fields shown in a view command in the API are specially defined as API defaults and are not subject to changes that may be reflected in the GUI.

Executing a Command

The command runner executes the command and passes it to the integration point. The integration point runs the command and returns a response, which contains all information related to the execution of the command. The response contents can then be accessed or another command can be run. Each command execution generates a new response. For more information on accessing response contents, see “Reading Responses” on page 26.

If a command specified for the Java API ends in error, a command level exception is thrown and should be caught. For more information, see “Catching Exceptions” on page 35.

Reading Responses

A *response* contains all of the output from executing a command. All commands communicate using the same response type, but use the response differently depending on the type of command and what it needs to communicate. This section describes the response for creation, action, and view command types.

NOTE Often it is useful to know the host, port, and user that a command used to connect to the Integrity Server. The MKS API has functions and methods to return this information. For more information, see the MKS Java API documentation or the MKS ANSI C API documentation.

Sample integrations that illustrate each of these command types are available on the MKS Developer Zone page of the MKS Web site (www.mks.com/devzone).

To find any errors that occurred as part of the response for a command specified for the C API, you need to search for them explicitly. For more information, see “C API Application Errors” on page 39.

Creation Commands

Creation commands create something new, for example, an issue. The command names usually start with the word “create”. The response for a creation command communicates the item that was created or what errors prevented the item from being created.

If the command ran successfully, the result contains information about what was created. The result message is the same text shown by the CLI, for example, “Created issue 123”.

If you want to continue working with the created item, you do not need this message; you only need the ID of what was created. The following examples show how to retrieve the item ID from the result.

- **Java Example: Retrieving the item ID**

```
Response response = cmdRunner.execute(myCreationCmd);
String newId =
    response.getResult().getPrimaryValue().getId();
```

- **C Example: Retrieving the Item ID**

```
response = mksCmdRunnerExecCmd(cr, cmd, NO_INTERIM);
result = mksResponseGetResult(response);
item = mksResultGetPrimaryValue(result);
wchar_t newId(256);
rtn = mksItemGetId(item, newID, sizeof(newId));
```

Action Commands

Action commands perform some action within the application, for example, editing, deleting, checking out, checking in, or resynchronizing. The response for an action command communicates what item the action was attempted on, and the result of the requested action, which can be either the change that occurred or an error that occurred while trying to change the item.

Unlike creation commands, which only create a single item, action commands can work with multiple items. The response for all action commands contains result and exception information for each item that was acted on. This allows for exact reporting of the item acted on, and the result or exception from that item. If an exception is present on any item, the command itself reports the failure with an exception of its own.

If the command runs successfully on a single item, you may just need to know from the response that no errors occurred. Make sure you know what the command reports before making this assumption. In other cases, you may need to retrieve more information from the response.

Retrieving Work Items

Work items indicate the items that the command processed. For example, running **si resync** without specifying a selection resynchronizes all members in a Sandbox and reports on each. The members can be retrieved from the command response by their work item ID, or through iteration as in the following examples.

- **Java Example: Retrieving Work Items**

```
Response response = cmdRunner.execute(myActionCommand);
for (WorkItemIterator i = response.getWorkItems();
    i.hasNext();) {
    WorkItem wi = i.next();

    //do something with the work item
}
```

■ C Example: Retrieving Work Items

```
response = mksCmdRunnerExecCmd(cr, cmd, NO_INTERIM);
workItem = mksResponseGetFirstWorkItem(response);

/* Do something with the first work item. */

while ((workItem = mksResponseGetNextWorkItem()) != NULL) {
    /* Do something with the work item. */
}
```

Each retrieved work item contains an ID, a context (if needed), a display string, a model type (for example, state, issue), and, optionally, a result.

Retrieving Work Item Results

Each work item optionally contains information about the action taken. For example, if you run **si resync** and there is nothing to resynchronize for a particular member, there is no result for that work item, indicating that no action was taken. If an action was taken, there is a result. In the case of **si resync**, the result is the revision that was checked out. The following examples show how to retrieve result information for a work item.

■ Java Example: Retrieving Work Item Results

```
String newRevisionId =
    workItem.getResult().getPrimaryValue().getId();
```

■ C Example: Retrieving Work Item Results

```
result = mksWorkItemGetResult(workItem);
item = mksResultGetPrimaryValue(result);
newId = (wchar_t *) malloc(256*sizeof(wchar_t));
rtn = mksItemGetId(item, newId, 256);
```

View Commands

View commands provide access to the current state of information within an application. Typically, the command names are either the name of the item type to be viewed (for example, **issues**, **sandboxes**), or they start with the word “view”. The response for a view command communicates the content of the item being viewed, no matter how complex.

Like action commands, view commands always return work items, command exceptions, and work item exceptions. Unlike action commands, a view command does not need to report the results of an action.

Retrieving View Information

Work items can be retrieved from the response for view commands in the same way as they are for action commands.

Once you have identified the work items, you can access their content. All work item content is contained in fields. You can access the information in the fields without parsing. The fields are either the basic data types (for example, string, Boolean) or contain a list of basic data

type values, items, or a list of items (a work item is a type of item). These items can, in turn, contain fields, which can contain items or a list of items. In this way, the response for a view command becomes a hierarchy of information that can be accessed easily.

To access view information from the response, you need to get the field from the item then read its value. If the field contains another item, you can get that item's ID, context, or model type, or any of its fields, and so on. You can use an iterator over the fields. The field value is null with a data type of null if the contents are not set within the application.

The following examples show how to retrieve a field value from a sample MKS Integrity issue.

- **Java Example 1: Retrieving the created date for the issue**

```
Date myDate = workItem.getField("Created Date").getDateTime();
```

In addition to retrieving the field as the matching data type as in this example, you can use the `getValueAsString()` method to return any data type as a string as in Java Example 2.

- **Java Example 2: Retrieving the created date for the issue as a string**

```
String myDateString = workItem.getField(
    "Created Date").getValueAsString();
```

For the basic data types, using the `getValueAsString()` method just returns the string representation; for fields that are items, it returns the item ID. See the MKS Java API Documentation for details of the behavior of this method.

- **Java Example 3: Retrieving the issue state**

```
String state = workItem.getField("State").getValueAsString();
```

- **Java Example 4: Retrieving the assigned user for the issue**

```
Item user = (Item)workItem.getField("Assigned User").getItem();
String fullname = user.getField("fullname").getValueAsString();
```

- **Java Example 5: Retrieving the list of issue relationships**

```
Field related = workItem.getField("MKSIssueRelationships");
Item first = (Item)related.getList().get(0);
String firstId = first.getId();
String firstState = first.getField("State").getValueAsString();
```

For fields that contain a list of values or a list of items, you can either extract a known element, as in this example, or you can iterate over the full list of items.

- **C Example 1: Retrieving the created date for the issue**

```
time_t myDate;
mksrtn rtn;
```

```
field = mksWorkItemGetField(workItem, "Created Date");
rtn = mksFieldGetDateTimeValue(field, &myDate);
```

In addition to retrieving the field as the matching data type as in this example, you can use the `mksFieldGetValueAsString()` function to return any data type as a string as in C Example 2.

■ **C Example 2: Retrieving the created date for the issue as a string**

```
wchar_t myDateString[256];
mksrtn rtn;

field = mksWorkItemGetField(workItem, "Created Date");
rtn = mksFieldGetValueAsString(field, myDateString,
    sizeof(myDateString));
```

For the basic data types, using the `mksFieldGetValueAsString()` function just returns the string representation; for fields that are items, it returns the item ID. See the MKS ANSI C documentation for details of the behavior of this function.

■ **C Example 3: Retrieving the issue state**

```
wchar_t state[256];
mksrtn rtn;

field = mksWorkItemGetField(workItem, "State");
rtn = mksFieldGetValueAsString(field, state, sizeof(state));
```

■ **C Example 4: Retrieving the assigned user for the issue**

```
wchar_t fullname[256];
mksItem user;
mksrtn rtn;

field = mksWorkItemGetField(workItem, "Assigned User");
rtn = mksFieldGetItemValue(field, &user);

field = mksItemGetField(item, "fullname");
rtn = mksFieldGetValueAsString(field, fullname, sizeof(fullname));
```

■ **C Example 5: Retrieving the list of issue relationships**

```
wchar_t firstId[256]; firstState[256];
mksItem first;
mksItemList itemList;
mksrtn rtn;

field = mksWorkItemGetField(workItem, "MKSIssueRelationships");
rtn = mksFieldGetItemListValue(field, &itemList);

first = mksItemListGetFirst(itemList);
rtn = mksItemGetId(first, firstId, sizeof(firstId));
```

```
field = mksItemGetField(first, "State");
rtn = mksFieldGetValueAsString(field, firstState,
    sizeof(firstState));
```

For fields that contain a list of values or a list of items, you can either extract a known element as in this example, or you can iterate over the full list of items.

Command Response Summary

The following table shows the generic response information for typical command executions.

Command Type	Response Pattern	Notes
Creation command - successful	<ul style="list-style-type: none"> ■ Application and command ■ Exit code of 0 ■ Result with created item ID and optional context 	Sample commands: <ul style="list-style-type: none"> ■ im createissue ■ si createsandbox
Action command - successful	<ul style="list-style-type: none"> ■ Application and command ■ Exit code of 0 ■ Work items with type and ID <ul style="list-style-type: none"> • Results with affected item ID and optional context 	Sample commands: <ul style="list-style-type: none"> ■ si co ■ si ci
View command - successful	<ul style="list-style-type: none"> ■ Application and command ■ Exit code of 0 ■ Work items with type and ID <ul style="list-style-type: none"> • Fields • Item Lists <ul style="list-style-type: none"> • Items with ID <ul style="list-style-type: none"> • Fields • Item Lists 	Sample commands: <ul style="list-style-type: none"> ■ im viewissue ■ im users The structure repeats as necessary to represent the view's data.
Command failed	<ul style="list-style-type: none"> ■ Application and command ■ Exit code of non 0 ■ API exception 	This applies to all types of commands.
Work item error	<ul style="list-style-type: none"> ■ Application and command ■ API exception ■ Exit code of non 0 ■ Work items with type and ID <ul style="list-style-type: none"> • API exception 	This applies to action and view type commands. The exception is on the work item that encountered the error. Work items that did not encounter an error have work item contents.

Viewing Responses With the API Viewer Utility

The response contents for each command depend on the command type and the specific command options used. MKS provides a utility that allows you to run any command on an Integrity Client or MKS Integrity Server. The complete contents of the response is returned to help you understand the API and how to read a particular response.

It is highly recommended that this utility be used to run any commands you use in an integration to familiarize yourself with their response and behavior through the API.

Running the API Viewer Utility for Java

To run the API viewer utility for the Java API, consult the MKS Java API documentation for `com.mks.api.util.APIViewer`. Some notable Java property settings include:

- `com.mks.api.response.showconnection=true` returns the host, port, and user of the command used to connect to the Integrity Server.
- `com.mks.api.xml.output=true` displays all command output in XML format.

Running the command

```
im issues "--fields=Summary,Type,Created Date" 13 14
```

through the MKS API Viewer could produce the following output:

```
Response:
  App. Name      = im
  Command Name   = issues
  Work Item:
    Id           = 13
    Context      = null
    Model Type   = im.Issue
    Field:
      Name       = Summary
      Data Type  = java.lang.String
      Value      = Table too large, must be
                  reworked.
    Field:
      Name       = Type
      Data Type  = com.mks.api.response.Item
      Item:
        Id       = Bug
        Context  = null
        Model Type = im.Type
    Field:
      Name       = Created Date
      Data Type  = java.util.Date
      Value      = Fri Dec 05 10:33:24 CST 2003
  Work Item
    Id           = 14
    Context      = null
    Model Type   = im.Issue
```



```

Field:
  Name      = Summary
  Data Type = java.lang.String
  Value     = The panel isn't leaving enough
              room for the buttons.

Field:
  Name      = Type
  Data Type = com.mks.api.response.Item
  Item:
    Id       = Bug
    Context  = null
    Model Type = im.Type

Field:
  Name      = Created Date
  Data Type = java.util.Date
  Value     = Wed Feb 04 10:33:43 CST 2004

Exit Code = 0

```

Running the API Viewer Utility for the C API

To run the API viewer utility for the C API, run the `mksAPIViewer` application located in the `<Integrity client install directory>\bin` directory. To view the usage commands, type `mksAPIViewer` in a command prompt or session window. Some notable options include:

- `--showconnection` returns the host, port, and user of the command used to connect to the Integrity Server.
- `--xml` displays all command output in XML format.

Running the command

```
im issues "--fields=Summary,Type,Created Date" 13 14
```

through the MKS API viewer could produce the following output:

```

Response:
  App. Name      = im
  Command Name   = issues
  Work Item:
    Id           = 13
    Context      = NULL
    Model Type   = im.Issue
  Field:
    Name         = Summary
    Data Type    = wchar_t *
    Value        = Table too large, must be
                  reworked.

  Field:
    Name         = Type
    Data Type    = mksItem
  Item:
    Id           = Bug

```

```

        Context      = NULL
        Model Type = im.Type
    Field:
        Name        = Created Date
        Data Type = time_t
        Value       = Fri Dec 05 10:33:24 CST 2003
Work Item
    Id          = 14
    Context     = NULL
    Model Type = im.Issue
    Field:
        Name        = Summary
        Data Type = wchar_t *
        Value       = The panel isn't leaving enough
                    room for the buttons.
    Field:
        Name        = Type
        Data Type = mksItem
    Item:
        Id          = Bug
        Context     = NULL
        Model Type = im.Type
    Field:
        Name        = Created Date
        Data Type = time_t
        Value       = Wed Feb 04 10:33:43 CST 2004
Exit Code = 0

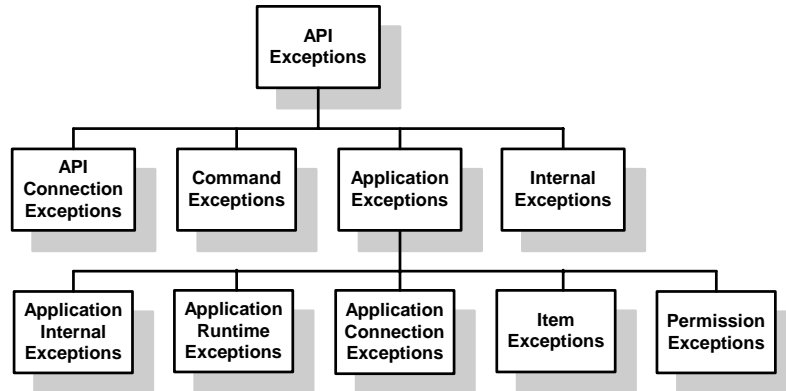
```

Java API Exceptions

This section describes the exceptions that can be thrown while using the MKS Java API.

MKS Java API exceptions use a generic model, with classes for each general type of exception. The exception classes represent the following types of exceptions:

- all exceptions associated with an API connection
- all exceptions associated with a command definition
- all specific application exceptions and their attributes



Catching Exceptions

If a command ends in error, a command level exception is thrown and should be caught. The exception class identifies the type of exception. The specific exception condition is identified by the exception ID, which is accessed by using the `getExceptionId()` method on the `APIException`.

Specific exceptions can contain additional information that can further identify the nature of the error. You need to read the response to get this information. The final command exit code can also be accessed from the response.

Use the `getResponse()` method on the caught exception as shown in the following example:

```

Response response = null;
try {
    response = cmdRunner.execute(myCmd);
    // read response
}

catch (APIException ae) {
    response = ae.getResponse();
}
  
```

Even when a command fails, it may have completed some work. For information on how to find out what work may have been successfully completed, see “Retrieving Work Items” on page 27.

Use the API Viewer utility to see the information returned for specific error conditions. For more information on using the API Viewer, see “Viewing Responses With the API Viewer Utility” on page 32.

If you are using interim responses, commands level exceptions are not thrown when executing a command. For more information, see “Using Interim Responses” on page 44.

Work Item Exceptions

Work item exceptions are thrown when retrieving a work item from a response using the `WorkItemIterator`. When a work item exception is thrown, all command output is still available in the response. You can determine the work item you were attempting to retrieve when the exception was thrown through the `WorkItemIterator.getLast()` method. If you are not using the `WorkItemIterator`, use the `getAPIException()` method on a work item to see if there is an error associated with the item.

The exception class identifies the general type of exception. To get a more specific exception, you need to retrieve the exception ID using the `getExceptionId()` method. Specific exceptions can contain additional information on the nature of the error.

If the error occurs as a result of anything other than executing a command on the Integrity Client or Integrity Server, there is no exception ID.

API Connection Exceptions

API connection exceptions occur when a connection fails between the API and the integration point. The following table gives some examples of connection exceptions and the reason why they occur.

Exception Class	Reason for Exception
<code>InvalidHostException</code>	API cannot connect to configured integration point. There is no server/client running at the configured address for integration point.
<code>InvalidIntegrationPointException</code>	API can connect to an address but the integration point details are invalid, for example, invalid host name or invalid port.
<code>IncompatibleVersionException</code>	Integration point does not support version of API that is attempting to communicate with it.

Command Exceptions

Command exceptions occur when a command does not execute.

NOTE More complex command specifications (for example, filter specifications, query specifications, rule specifications) may fail due to MKS Integrity or MKS Source validation errors, but still be reported as command exceptions.

The following table gives some examples of command exceptions and the reason why they occur.

Exception Class	Reason for Exception
<code>CommandAlreadyRunningException</code>	Command runner already in use and cannot be used to run another command without first interrupting it.
<code>CommandCancelledException</code>	Command has been cancelled due to a failed pre-condition, for example, a user pressed Cancel button when asked for settings.
<code>InvalidCommandOptionException</code>	Invalid command option used or command option missing.
<code>InvalidCommandRunnerStateException</code>	Command runner cannot be used to execute commands, for example, command runner has been released.
<code>InvalidCommandSelectionException</code>	Invalid command selection used or command selection missing.
<code>UnknownCommandException</code>	Command unknown.
<code>UnsupportedFunctionException</code>	Error occurred while executing a command, for example, unsupported command feature such as <code>--nobatch</code> used.
<code>UnsupportedVersionException</code>	API version not compatible with integration point.

Application Exceptions

Application exceptions occur when there is a processing or validation error within MKS Integrity or MKS Source.

The application exception classes represent only the generic application exceptions. There are many more specific exceptions that are possible within each generic application exception. Each specific exception is mapped to an exported exception name used for API exception reporting. The specific exception that occurred can be identified using the `getExceptionId()` method.

The three main types of API application exceptions are:

- application connection exceptions
- item exceptions
- permission exceptions

The following table lists some general API application exceptions.

Exception Class	Reason for Exception
<code>ApplicationInternalException</code>	Internal error occurred on integration point while executing command.
<code>ApplicationRuntimeException</code>	Error occurred while integration point running command because of unanticipated application problem.
<code>UnsupportedApplicationException</code>	Application passed in with command unknown/invalid.

Application Connection Exceptions

Application connection exceptions occur when there is an error while initiating a connection to the MKS Integrity Server.

The following table lists the application connection exceptions and the reason why they occur.

Exception Class	Reason for Exception
<code>IncompatibleVersionException</code>	Valid integration point client cannot connect to server because server does not support client's version.
<code>NoCredentialsException</code>	User's credentials missing.

Item Exceptions

Item exceptions occur when there is an error processing an item while executing a command.

The following table lists the application item exceptions and the reason why they occur.

Exception Class	Reason for Exception
<code>InvalidItemException</code>	Item invalid.
<code>ItemNotFoundException</code>	Item does not exist, for example, member you are trying to drop from Sandbox does not exist.
<code>ItemAlreadyExistsException</code>	Item is duplicate, for example, member you are trying to add to Sandbox already added.
<code>ItemModificationException</code>	Item cannot be modified, for example, MKS Integrity field you are trying to change is read-only.

Permission Exceptions

Permission exceptions occur when a command cannot be executed because the user did not have the right permissions.

Currently the permission exceptions are only guaranteed to contain a message. They do not necessarily indicate the object or the permission being checked.

Internal Exceptions

Internal exceptions occur when there is an internal error within the integration point. The following table lists the internal exception and the reason why it occurs.

Exception Class	Reason for Exception
UnknownException	Unknown error occurred while executing command.

Integrity Client Launch Exceptions

Integrity Client launch exceptions occur when the Integrity Client is prevented from being started by user settings, for example, when the native path is not set to include the client install.

Interrupted Exceptions

Interrupted exceptions occur when you try to access a response when the command runner was interrupted before the command completed.

C API Errors

The error handling approach for the C API differs from normal C error handling. Normally the `errno` global variable is used to hold the last known error code, and the `strerror()` function is used to provide an error message. To support pointers to multiple threads and ensure thread safety, the `mksrtn` variable holds the last known error code, and the `mksAPIStrError()` function provides an error message.

There are two types of errors that can be generated by the C API:

- application errors that are part of the response
- internal API errors that are not part of the response

C API Application Errors

To find any errors that occurred as part of the response, you need to search for them explicitly, for example:

```
mksrtn rtn = MKS_SUCCESS;

rtn = mksItemGetId(item, buf, sizeof(buf));
if (rtn != MKS_SUCCESS) {
    /* Handle the error here. */
}
```

Some C API functions return `NULL` if an error has occurred, rather than an error code. For these functions, you need to retrieve the error code in a different way.

For functions that return pointers to structs, like `mksCmdRunnerGetSession()`, `mksSessionGetIntegrationPoint()`, `mksCmdRunnerExecCmd()`, and `mksCmdRunnerExecArr`, use the `mksGetError()` function to retrieve the last generated error, and the `mksAPIStrError()` function to retrieve the error message for the related error code, for example:

```
mksResponse response = NULL;
response = mksCmdRunnerExecCmd(cr, cmd, NO_INTERIM);
if (response == NULL) {
    rtn = mksGetError();
    /* Handle the error here. */
}
```

NOTE The return value of `mksGetError()` is only updated after making a call to a function that does not return `mkstrn` or `void`. For example, calling `mksResultGetMessage()` after calling `mksResultGetField()` does not reset the return value of `mksGetError()` as was set by the `mksResultGetField()` function.

For the remaining functions that return `NULL`, use the `mksGetError()` function to retrieve the `mkstrtn` error code value, and the `mksAPIStrError()` function to retrieve the string associated with the error.

Application Error Codes

The application error codes represent only the generic application exceptions. There are many more specific exceptions that are possible within each generic application exception. Each specific exception is mapped to an exported exception name used for API exception reporting. The specific exception that occurred can be identified using the `mksAPIExceptionGetId()` method.

The following tables list the application error codes that can occur, what they mean, and the reason why they occur.

Error Code	Description	Reason for Error
300	General application exception	Processing or validation error within MKS Integrity or MKS Source.
301	Unsupported application	Application passed in with command is unknown/invalid.
302	Internal application error	Internal error occurred on integration point while executing command.
303	No such element error	Attempted to access element that does not exist, for example, field.

Error Code	Description	Reason for Error
310	Application connection failure	Error while initiating connection to MKS Integrity Server.
311	Incompatible version	Valid integration point client cannot connect to server because server does not support client's version.
312	No credentials	User's credentials missing.
320	Application runtime error	Error occurred while integration point running command because of unanticipated application problem.
321	Application out of memory error	Integration point ran out of memory while executing command.
330	Generic item exceptions	Error processing an item while executing command.
331	Invalid item	Item invalid.
332	Item already exists	Item is duplicate, for example, member you are trying to add to Sandbox already added.
333	Item modification exception	Item cannot be modified, for example, MKS Integrity field you are trying to change is read-only.
334	Item not found	Item does not exist, for example, member you are trying to drop from Sandbox does not exist.
340	Permission failure	Permission exceptions occur when command cannot be executed because user did not have right permissions.

Internal Error Codes

The following tables list the internal error codes that can occur, what they mean, and the reason why they occur.

Error Code	Description	Reason for Error
200	Out of memory error	Not enough memory allocated.
201	Interrupted	Cannot access a response because command runner interrupted before command completed.
202	Integrity client launch error	Integrity Client cannot be started because of user settings, for example, native path not set to include client install.
203	Invalid parameter - NULL	NULL parameter passed to function expecting valid value.

Error Code	Description	Reason for Error
204	Invalid parameter - NULL list	NULL parameter list passed to function expecting valid parameter list.
205	Insufficient buffer size	Buffer too small to accept data.
206	Invalid data type	Integration attempting to cast one data type into another incompatible one.
207	End of list	Integration attempting to access more list items available.
210	Command already running	Command runner already in use and cannot be used to run another command without first interrupting it.
211	Invalid command runner state	Command runner cannot be used to execute commands, for example, command runner has been released.
220	API connection failure	Connection failure occurred when trying to connect with integration point.
221	Invalid host	API cannot connect to configured integration point. No server/client running at the configured address for integration point.
222	Invalid integration point	API can connect to address but integration point details are invalid, for example, invalid host name or invalid port.
223	Unsupported API version	API version not compatible with integration point.
230	Generic API communications error	Communication between integration and integration point broken.

Command Variations

The following variations can be used when running commands:

- “Using the Graphical User Interface” on page 43
- “Using Interim Responses” on page 44
- “Using Impersonation” on page 45
- “Using File References” on page 46
- “Using Uncertified Commands” on page 47

Using the Graphical User Interface

If you are integrating with the Integrity Client, you may want your users to interact with the client through the GUI. The GUI interaction is the same whether you use a standard session or the common session (for more information, see “Sessions” on page 22).

There are several ways to interact through the GUI:

- full GUI
- GUI for prompting
- GUI for status reporting

NOTE If your integration runs as an NT service and automatically starts the Integrity Client on the same system, that client is not able to perform GUI functions since it has no ability to display.

GUI for Status Reporting

The response for an API command gets some of its content from status information generated by the command. The status identifies the work items that have been processed, any results generated from processing the work items or by the command itself, and any errors that occurred. The status does not include any of the informational output from view commands.

Status reporting is controlled through the `--status` command option. You can turn off status reporting (`--status=none`) or you can direct status reporting to the GUI (`--status=gui`).

If the status is redirected to the GUI, then all the status information is presented directly to the user through the GUI. This option is especially helpful for long running commands (for example, resyncing a Sandbox) to let the user know that something is happening.

If the status is redirected to the GUI, the status information is not included in the API response. The API response does, however, always contain the final exit code from the command, as well as any final exception if the command failed.

TIP If all you need in your response is the exit code, using `--status=none` is a good way to reduce the size of the response on action commands.

GUI for Prompting

GUI prompting of command settings can be added to any command run through the API by specifying the option `--settingsUI=gui`. This causes the command settings dialog box to display before the command executes, enabling you to see the prompts and specify your settings.

When you use this option, command status information is presented to the user through the GUI. Unlike the `--status=gui` option, status information is also included in the API response.

Full GUI

The full GUI is invoked when the option `--gui` is specified on a command. Using this option redirects all information from a view command to the GUI. When you use this setting, the API response for view commands only contains an exit code and any final command exception.

Using this option also enables GUI prompting of command settings for all types of commands.

Using Interim Responses

When a command returns a large amount of data or takes a long time between returning the first piece of information and the last, you can begin reading the initial response information before the command has completed running. For example, you might want to use an interim response when running a query in MKS Integrity that returns details on thousands of issues. If you use an interim response you can read the first issue immediately.

If you use an interim response on a command, it returns an empty response immediately, and you can begin trying to read work items. This causes your integration program to wait until the first work item is returned from the command. You should not read command level information (for example, command level results, exceptions, exit codes) until the command completes. Since command level information is initially unavailable, any Java API command exception that occurs is recorded in the response, but is not thrown when executing the command. If the work item results in an error, the appropriate exception is thrown.

Interim responses can be cached or not. Not caching improves memory management because the response is not populated with all the work items. Once you have read all of the work items, the response is empty.

CAUTION If you are not caching and you request command level information, such as exit code, all of the unread work items are discarded, even if they have not all been read.

If performance is more important than memory management, you can use the interim response for quick reading and enable caching of returned work items. This allows you to read them as they are received, but also builds up a complete response containing all of the returned work items.

To use interim responses for the Java API

- 1 Execute the command using the `executeWithInterim()` method.
- 2 Start reading work items from the response using the `WorkItemIterator`.

To use interim responses for the C API

- 1 Execute `mksCmdRunnerExecArr()` or `mksCmdRunnerExecCmd()` with one of the `INTERIM_NO_CACHE` or `INTERIM_CACHE` enum options.
- 2 Start reading work items from the response using the `MKSResponseGetFirstWorkItem()` and `MKSResponseGetNextWorkItem` functions.

Using Impersonation

In a server to server integration, where the server is doing work for multiple users, the integration is unable to supply passwords for every user to the MKS Integrity Server because it usually does not have access to them. *Impersonation* provides a controlled way for an integration to work on behalf of another user. Before an impersonation can be performed, an administrator must set up who can be impersonated and who is allowed to use the impersonation. For details of how to set up impersonations, see “Setting Up Impersonations” on page 97.

You must still supply a default user and password for the impersonating user. In most cases, the default user and password are the same for all impersonated users within an integration.

NOTE Each impersonated user consumes a license as if the impersonated user had logged on.

■ Java Example: Using Impersonation

```
cmdRunner.setDefaultHostname("myserver");
cmdRunner.setDefaultPort(7001);
cmdRunner.setDefaultUsername("IntegrationUser");
cmdRunner.setDefaultPassword("IntegrationUserSecret");
cmdRunner.setDefaultImpersonationUser("real user");
```

■ C Example: Using Impersonation

```
rtn = mksCmdRunnerSetDefaultHostname(cmdRunner, "myserver");
rtn = mksCmdRunnerSetDefaultPort(cmdRunner, 7001);
rtn = mksCmdRunnerSetDefaultUsername(cmdRunner, "IntegrationUser");
rtn = mksCmdRunnerSetDefaultPassword(cmdRunner, "Secret");
rtn = mksCmdRunnerSetDefaultImpersonationUser(cmdRunner,
"real user");
```

Using File References

Some application commands reference the file system either as an input or output, for example:

- MKS Source Sandbox commands
- MKS Source check out command
- MKS Source check in command
- MKS Source command that provide bulk data transfer
- MKS Integrity commands that specify an image or issue attachment

These commands still interact with the file system when run through the API. There are three things to keep in mind when working with commands with file system references:

- integration point
- current working directory
- location of the API

Integration Point

File system references are applied at the integration point not at the integration location. When constructing a command to send to a remote integration point, the file system references must be valid on the remote system, which is where the files are read or written. All file system references are valid when using an Integrity Client as an integration point. All file references on the Integrity Server are restricted to `server/temp` in the server install directory for security reasons.

If you need complete file system access, but want to integrate your server application directly to an Integrity Server, consider the scenario described in “Integrating With the Integrity Client as Server” on page 13.

Current Working Directory

Several commands have implicit logic to derive settings that are not specified in the command by searching the system, starting with the current working directory. When using the API, if no current working directory (`--cwd`) setting has been set on the command, the directory that the current integration application started in is used as the current working directory. For this reason, you must either explicitly include all settings for files (for example, the Sandbox or project name) or determine what the current working directory should be and pass it as the `--cwd` setting for all commands that use it.

Location of the API

The **si projectadd**, **si projectco**, and **si projectci** commands enable you to transfer bulk data by specifying a source or output file. If you use these commands through the API and want to transfer bulk data to the API's file system rather than the Integration Point's file system, you must create a special command option. For the MKS Java API, you create this option using the `com.mks.api.FileOption` class. For the MKS C API, you create this option using the `mksOptionListAddFile` method. When you use this option, any file locations that you specify are relative to the location of the API.

For more information on Java API classes or C API methods, see the MKS Java API documentation or MKS C API documentation, both of which can be accessed through the Integrity Server Web page. For more information on the bulk data commands, see the **si projectadd**, **si projectco**, and **si projectci** commands in the *MKS Source 2006 CLI Reference Guide* and the **im extractattachments** command in the *MKS Integrity 2006 CLI Reference Guide*.

Using Uncertified Commands

You can use uncertified commands to initiate commands and views that use the GUI. Much of the response content for uncertified commands is missing or incomplete. Some uncertified commands throw command level exceptions.

Uncertified command responses contain:

- the application and command
- the exit code (same as the CLI)
- exceptions containing only a message, no ID or fields
- results containing only a message

Any data returned by uncertified commands, and even the entire response structure, is subject to change when the commands are certified, and should not be used.

For a list of certified commands, see “API Certified Commands” on page 59.

Best Practices

This section describes some of the recommended practices for using the API:

- “Copy the API” on page 48
- “Managing Users” on page 48
- “Memory Management Tips” on page 53
- “Use Facade Patterns” on page 54

- “Converting MKS Source Scripts to API Programs” on page 54
- “Accessing Item Fields” on page 55

Copy the API

If you are using the API, when you run your integration you should use the same version of the API that you used to compile your integration. It should be copied and bundled with the integration. If you rely on the API that is installed on the client or server, when the version number changes (that is, when the client or server is upgraded), the next time you try to run your integration you may have binary compatibility issues. For more information, see “Java and C API Compatibility” on page 98.

NOTE You should also keep a copy of the matching MKS API documentation from the MKS Integrity Server.

As an alternative to keeping a copy of the API, you could make sure that a newly compiled version of your integration is used with the upgraded API files.

Managing Users

An integration uses sessions differently depending on the number of users it is managing. To best understand the information in this section, you should be familiar with the information outlined in “Sessions” on page 22.

Single User

A *single user* session is the scenario that a lot of scripts are written for: run a sequence of commands for a single user. A single user session can be used for a client or server integration point. The following examples show how to create a single user session for both integration points.

NOTE You specify the user and password in two places. The session is used to validate that you have authority to create a session on the server integration point. The command runner defaults identify the user to run the command as. A different user and password could be used for each of these.

Java Example: Client Integration

```
IntegrationPointFactory ipf =
    IntegrationPointFactory.getInstance();
CmdRunner cmdRunner =
    ipf.createLocalIntegrationPoint()
        .getCommonSession()
        .createCmdRunner();
// Set only to override client defaults
```



```
cmdRunner.setDefaultHostname("myserver");
cmdRunner.setDefaultPort(7001);
cmdRunner.setDefaultUsername("IntegrationUser");
cmdRunner.setDefaultPassword("secret");
```

Java Example: Server Integration

```
IntegrationPointFactory ipf =
    IntegrationPointFactory.getInstance();
IntegrationPoint intPt =
    ipf.createIntegrationPoint("myserver", 7001);
CmdRunner cmdRunner =
    intPt.createSession("IntegrationUser", "secret")
        .createCmdRunner();
// Set command defaults
cmdRunner.setDefaultHostname("myserver");
cmdRunner.setDefaultPort(7001);
cmdRunner.setDefaultUsername("IntegrationUser");
cmdRunner.setDefaultPassword("secret");
```

C Example: Client Integration

```
mksIntegrationPoint ip = NULL;
mksSession session = NULL;
mksCmdRunner cr = NULL;
mksCommand = NULL;
mksrtn rtn = MKS_SUCCESS;
rtn = mksAPIInitialize(NULL);
rtn = mksCreateLocalIntegrationPoint(&ip, 1);
rtn = mksGetCommonSession(&session, ip);
rtn = mksCreateCmdRunner(&cr, session);
/* Setup defaults for the mksCmdRunner.*/
rtn = mksCmdRunnerSetDefaultHostname(cr, "myserver");
rtn = mksCmdRunnerSetDefaultPort(cr, 7001);
rtn = mksCmdRunnerSetDefaultUsername(cr, "IntegrationUser");
rtn = mksCmdRunnerSetDefaultPassword(cr, "secret");
```

C Example: Server Integration

```
mksIntegrationPoint ip = NULL;
mksSession session = NULL;
mksCmdRunner cr = NULL;
mksCommand cmd = NULL;
mksrtn rtn = MKS_SUCCESS;
rtn = mksAPIInitialize(NULL);
rtn = mksCreateIntegrationPoint(&ip, "myserver", 7001, 0);
rtn = mksGetSession(&session, ip, "IntegrationUser", "secret");
rtn = mksCreateCmdRunner(&cr, session);
/* Setup our defaults for the mksCmdRunner. */
rtn = mksCmdRunnerSetDefaultHostname(cr, "myserver");
rtn = mksCmdRunnerSetDefaultPort(cr, 7001);
```

```

rtn = mksCmdRunnerSetDefaultUsername(cr, "IntegrationUser");
rtn = mksCmdRunnerSetDefaultPassword(cr, "secret");

```

Multiple Users

When there are multiple users, you need to make sure that one user's actions do not have an undesirable effect on another user's actions. You can achieve this by creating a session for each user or by sharing a session. Sharing a session allows you to manage your resources with greater efficiency.

When working with multiple users in a shared session, you may want to consider using impersonation so that the integration is not required to use each user's password. For more information, see "Using Impersonation" on page 45.

The following examples shows how to share a single session among multiple users. When combined with multi threading, this can allow an integration to have multiple threads, each servicing a user or multiple users sequentially.

Java Example: Sequential Users

```

IntegrationPointFactory ipf =
    IntegrationPointFactory.getInstance();
IntegrationPoint intPt =
    ipf.createIntegrationPoint("myserver",7001);
CmdRunner cmdRunner =
    intPt.createSession("IntegrationUser","secret")
        .createCmdRunner();
// Set command defaults
cmdRunner.setDefaultHostname("myserver");
cmdRunner.setDefaultPort(7001);
cmdRunner.setDefaultUsername("IntegrationUser");
cmdRunner.setDefaultPassword("secret");
// Run multiple commands
Response r = cmdRunner.execute(myCmd);
...
// Change User
cmdRunner.setDefaultUsername("AnotherUser");
cmdRunner.setDefaultPassword("theirSecret");
// Run multiple commands
r = cmdRunner.execute(myCmd);
...
[Repeat for as many users as necessary]

```

Java Example: Concurrent Users

```

// Setup a common session
IntegrationPointFactory ipf =
    IntegrationPointFactory.getInstance();
IntegrationPoint intPt =
    ipf.createIntegrationPoint("myserver",7001);
Session session =

```

```

        intPt.createSession("IntegrationUser", "secret");
    ...
    // When changing users
    // - share existing session
    // - create unique cmdRunner
    CmdRunner cmdRunner = session.createCmdRunner();
    cmdRunner.setDefaultHostname("myserver");
    cmdRunner.setDefaultPort(7001);
    cmdRunner.setDefaultUsername("IntegrationUser");
    cmdRunner.setDefaultPassword("secret");
    cmdRunner.setDefaultImpersonationUser(myUser);

```

C Example: Sequential Users

```

mksIntegrationPoint ip = NULL;
mksSession session = NULL;
mksCmdRunner cr = NULL;
mksCommand cmd = NULL;
mksResponse response = NULL;
mksrtn rtn = MKS_SUCCESS;

rtn = mksAPIInitialize(NULL);
rtn = mksCreateIntegrationPoint(&ip, "myserver", 7001, 0);
rtn = mksGetSession(&session, ip, "IntegrationUser", "secret");
rtn = mksCreateCmdRunner(&cr, session);

/* Setup defaults for the mksCmdRunner. */
rtn = mksCmdRunnerSetDefaultHostname(cr, "myserver");
rtn = mksCmdRunnerSetDefaultPort(cr, 7001);
rtn = mksCmdRunnerSetDefaultUsername(cr, "IntegrationUser");
rtn = mksCmdRunnerSetDefaultPassword(cr, "secret");

/* Run multiple commands */
response = mksCmdRunnerExecuteCmd(cr, myCmd, NO_INTERIM);
...
rtn = mksCmdRunnerSetDefaultUsername(cr, "AnotherUser");
rtn = mksCmdRunnerSetDefaultPassword(cr, "theirSecret");
/* Run multiple commands */
response = mksCmdRunnerExecuteCmd(cr, myCmd, NO_INTERIM);
...
[Repeat for as many users as necessary]

```

C Example: Concurrent Users

```

mksIntegrationPoint ip = NULL;
mksSession session = NULL;
mksCmdRunner cr1 = NULL;
mksCmdRunner cr2 = NULL;
mksCommand cmd = NULL;
mksResponse response = NULL;
mksrtn rtn = MKS_SUCCESS;

rtn = mksAPIInitialize(NULL);
rtn = mksCreateIntegrationPoint(&ip, "myserver", 7001, 0);

```

```

rtn = mksGetSession(&session, ip, "IntegrationUser", "secret");
...
/* When changing users:
   - share existing session
   - create a unique cmdRunner
*/
rtn = mksCreateCmdRunner(&cr, session);
rtn = mksCmdRunnerSetDefaultHostname(cr1, "myserver");
rtn = mksCmdRunnerSetDefaultPort(cr1, 7001);
rtn = mksCmdRunnerSetDefaultUsername(cr1, "IntegrationUser");
rtn = mksCmdRunnerSetDefaultPassword(cr1, "secret");

/* Run multiple commands */
response = mksCmdRunnerExecuteCmd(cr1, myCmd, NO_INTERIM);

```

Dedicated Sessions

If a session is dedicated to a single user, any actions one user performs cannot be affected by another user's actions through the session. If a session is used frequently, it should be pooled for re-use, rather than destroyed and re-created.

Java Example

```

// Setup integration point
IntegrationPointFactory ipf =
    IntegrationPointFactory.getInstance();
IntegrationPoint intPt =
    ipf.createIntegrationPoint("myserver", 7001);
...
// For Each User
// - create unique session
// - create unique cmdRunner
Session session =
    intPt.createSession("IntegrationUser", "secret");
CmdRunner cmdRunner = session.createCmdRunner();
cmdRunner.setDefaultHostname("myserver");
cmdRunner.setDefaultPort(7001);
cmdRunner.setDefaultUsername("IntegrationUser");
cmdRunner.setDefaultPassword("secret");
cmdRunner.setDefaultImpersonationUser(myUser);

```

C Example

```

mksIntegrationPoint ip = NULL;
mksSession session = NULL;
mksCmdRunner cr1 = NULL;
mksCmdRunner cr2 = NULL;
mksCommand cmd = NULL;
mksResponse response = NULL;
mksrtn rtn = MKS_SUCCESS;

```

```

rtn = mksAPIInitialize(NULL);
rtn = mksCreateIntegrationPoint(&ip, "myserver", 7001, 0);
...
/* For each user:
    - Create unique mksSession
    - Create unique mksCmdRunner
*/
rtn = mksGetSession(&session, ip, "IntegrationUser", "secret");

/* When changing users:
    - share existing session
    - create a unique cmdRunner
*/
rtn = mksCreateCmdRunner(&cr, session);
rtn = mksCmdRunnerSetDefaultHostname(cr1, "myserver");
rtn = mksCmdRunnerSetDefaultPort(cr1, 7001);
rtn = mksCmdRunnerSetDefaultUsername(cr1, "IntegrationUser");
rtn = mksCmdRunnerSetDefaultPassword(cr1, "secret");

/* Run multiple commands */
response = mksCmdRunnerExecuteCmd(cr1, myCmd, NO_INTERIM);

```

Memory Management Tips

You can use the following practices to manage your memory better within the integration and integration point.

- **Memory management within the integration**

If a command has the potential of returning a large number of work items or work items with large amounts of information, consider using the approach described in “Using Interim Responses” on page 44.

You should de-reference a response once it has been used so that it is available for garbage collection. You should also de-reference exceptions since they hold a reference to their associated response.

C API users must be sure to use the `mksAPITerminate()` function to release all the cached `mksIntegrationPoint` instances and close down the logger properly.

- **Memory management within the integration point**

Connections consume resources, whether they are established explicitly or as part of a command pre-condition. To manage your memory effectively, you should consider when to disconnect a connection. If a connection is used frequently, its session should be pooled for re-use.

In the same manner, when you are done with a session or command runner, you can release the integration point to free up any resources being used. Just de-referencing

these objects does not free up the resources. Releasing a session or command runner does not disconnect a client from a server.

In addition, remember to `execute` an `exit` to shut down clients that are automatically restarted.

- **Memory management within the response (C API only)**

The C API allows an `mksResponse` instance to be released. The `mksReleaseResponse()` function releases the **entire** response hierarchy. Since the response only contains references to the sub-structures, if you release the response you cannot re-use parts of the underlying hierarchy (unless you make a copy of it).

NOTE The response hierarchy of an `mksResponse` includes any `mksSubRoutine`, `mksWorkItem`, `mksItem`, `mksField`, `mksResult`, `mksAPIException`, `mksItemList` and `mksValueList` instances.

Use Facade Patterns

This practice only relates to the Java API. A *facade* is a way of providing a simplified, standardized interface to a large set of objects. If you write an integration that involves more than running a few commands, you should consider isolating the MKS API behind a facade. This involves creating a class or classes that hide the implementation of the function you need.

For example, if you needed to know the state of an issue, you could write a method such as `String getState(int issue)` that would return the state as a string when given an issue as an integer. This would enable you to use this function anywhere in your integration without needing to know about the workings of the API. When you need to make changes or go to a newer version of the API to use new features, your changes are isolated behind the facade and can be independently tested.

Depending on your needs, you can create a facade for each command, for all functions related to a single item type (for example, Sandboxes or issues), or for all the functions you are calling.

Converting MKS Source Scripts to API Programs

When converting MKS Source scripts to programs using the API, you often cannot take the exact commands being run, implement them in Java or C, and run them with the same results. The reason for this is that most MKS Source scripts interact with the file system in some way. You need to adjust the commands as described in the section “Using File References” on page 46.'

Accessing Item Fields

Fields on an item or work item can be of many basic types (for example, integer, Boolean, string) as well as some more complex types (for example, list, item, or item list). No matter what the field data type is, it is possible to retrieve the field as a string.

If you have full user name support enabled, the user is always returned as an item, and that item contains a field for the user's full name. For more information on full user name support, see the *MKS Integrity Server 2006 Administration Guide*. If all you want to do is determine the user's login ID, you can do this simply with the `getValueAsString()` method.

Retrieving Item Fields as Strings for the Java API

To retrieve item fields as string for the Java API, use the `getValueAsString()` method on the `Field` class.

For example, after running the command:

```
im issues --fields=Summary,Created By 1234
```

and getting the work item, the field values can be determined by:

```
String summary =
    workItem.getField("Summary")
        .getValueAsString();
String userlogin =
    workItem.getField("Created By")
        .getValueAsString();
```

For most data types, this method just returns the field value as a string. For fields that are item data types, this method returns the item ID. This saves the trouble of getting the item and then getting its ID. For fields that are list data types, this method returns the standard list contents available through the `java.util.List` interface. For item list data types, this method returns the standard Java Foundation Class list format of the IDs of the items in the list.

Retrieving Item Fields as Strings for the C API

To retrieve item fields as string for the C API, use the `mksFieldGetValueAsString()` function on the `Field` variable.

For example, after running the command:

```
im issues --fields=Summary,Created By 1234
```

and getting the work item, the field values can be determined by:

```
mksField field = NULL;
mksWorkItem wi = NULL;
wchar_t summary[1024];
wchar_t userlogin[1024];
```

```

mksrtn rtn;
...
field = mksWorkItemGetField(wi, "Summary");
rtn = mksFieldGetValueAsString(field, summary, sizeof(summary));
...
field = mksWorkItemGetField(wi, "Created By");
rtn = mksFieldGetValueAsString(field, userlogin, sizeof(userlogin));

```

For most data types this function just returns the field value as a string (`wchar_t *`). This function returns the ID for an `mksItem` value, a comma-separated list of IDs for an `mksItemList` value, or a comma-separated list of the elements for an `mksValueList` value. A `NULL` value returns an error code of `MKS_NULL_VALUE`.

Creating Programs for the C API

For all platforms, `mksapi.h` is the main include file, and `#include <mksapi.h>` includes all the files you need to compile.

Windows (Visual C++)

Add:

- `<Integrity Client>/include` as an additional include directory
- `<Integrity Client>/bin` as an additional library directory
- `mksapi1.lib` as an additional library to link

Solaris

You need:

- `-I<Integrity Client>/include` as a compiler flag
- `-L<Integrity Client>/lib -lmksapi -lapiblimp -lxml2 -lssl -lnspr4 -lcrypto -lpthread -lm -lsocket -lnsl` as linker flags

Linux

You need:

- `-I<Integrity Client>/include` as a compiler flag
- `-L<Integrity Client>/lib -lmksapi -lapiblimp -lxml2 -lssl -lnspr4 -lcrypto -lpthread -lz -lm` as a linker flag

HP/UX

You need:

- `-I<Integrity Client>/include` as a compiler flag

- `-L<Integrity Client>/lib -Wl,+s -lmksapi` as a linker flag

AIX

You need:

- `-I<Integrity Client>/include` as a compiler flag
- `-L<Integrity Client>/lib -lmksapi` as a linker flag

API Certified Commands

Command Work Items and Results

Every command and command option on the server/client can be called through the API, but only certified commands produce predictable results. This chapter lists all the certified commands. Any other commands are not guaranteed to work or to return predictable results. For more information, see “Using Uncertified Commands” on page 47.

For information on the commands and their options, see the *MKS Integrity 2006 CLI Reference Guide* and the *MKS Source 2006 CLI Reference Guide*.

For information on the custom change package commands and their options, see the *MKS Integrity Server 2006 Administration Guide*.

It is highly recommended that when you are designing integrations you use the API Viewer to become familiar with the expected output for API commands. For more information, see “Viewing Responses With the API Viewer Utility” on page 32.

Format for Command Information

Command information in this chapter is provided in the following format.

Command

Describes the command, including the command type (action, view, creation). For more information on command types, see “Reading Responses” on page 26.

Work Item

Describes the item that the command worked on, for example, a member.

Work Item Context

Describes the context of the work item, for example, the project for the member.

Work Item Result

Describes the result of the work item.

For action commands, list any potential fields for command results.

For view commands, lists any fields that are specifically added to the view by the API, or which are required to maintain structure on exposed field values. All other fields exposed for view commands are documented in the man pages for the commands and in the *MKS Source 2006 CLI Reference Guide* or *MKS Integrity 2006 CLI Reference Guide*.

It is highly recommended that you use the API Viewer to become familiar with the expected output for API commands. For more information, see “Viewing Responses With the API Viewer Utility” on page 32.

Command Result

Describes the result of the command.

Result Context

Describes the context of the result, for example, the change package type for the change package attribute.

NOTE No explicit order is implied by the sequence of the entries in any of the tables.

Certified Commands

MKS Integrity

- about
- columnsets
- connect
- copyissue
- cattributes
- centryattributes
- cps
- cptypes
- createcp
- createcpattribute
- createcpentry
- createcpentryattribute
- createcptype
- createissue
- deletecp
- deletecpattribute
- deletecpentryattribute
- deletecptype
- disconnect
- editcp
- editcpattribute
- editcpentry
- editcpentryattribute
- editcptype
- editissue
- extractattachments
- exit
- fields
- groups
- importissue
- types
- users
- issues
- projects
- queries
- servers
- states
- viewcolumnset
- viewcp
- viewcpattribute
- viewcpentryattribute
- viewcptype
- viewfield
- viewgroup
- viewissue
- viewproject
- viewquery
- viewstate
- viewtype
- viewuser

MKS Source

- about
- add
- addsubproject
- archiveinfo
- ci
- closecp
- co
- connect
- configuresubproject
- createcp
- createproject
- createsandbox
- createsubproject
- disconnect
- drop
- exit
- gui
- lock
- locks
- projectadd
- projectci
- projectco
- projectinfo
- rename
- resync
- revert
- sandboxinfo
- unlock
- viewhistory
- viewproject
- viewsandbox

Standard Data Types

The following Java data types are used for fields:

- `java.lang.Boolean`
- `java.util.Date`
- `java.lang.Double`
- `java.lang.Float`
- `java.lang.Integer`
- `java.lang.Long`
- `java.lang.String`
- `com.mks.api.response.Item`
- `com.mks.api.response.ItemList`
- `com.mks.api.response.ValueList`

The following C data types are used for fields:

- `NULL`
- `unsigned short (boolean)`
- `time_t (datetime)`
- `double`
- `float`
- `int`
- `long`
- `char * (string)`
- `wchar_t * (string)`
- `mksItem`
- `mksItemList`
- `mksValueList`

Unicode Support for C API

The C API has been designed to be fully Unicode through the use of the `wchar_t*` data type. There are, however, some exceptions:

- Full Unicode support for the C API is not supported on the HP-UX 11 platform due to HP-UX missing needed Unicode routines. However, HP-UX 11i (11.11) is supported.
- There is no Unicode support for C API commands that accept or return user names, passwords, host names, and file names. This includes the following functions:
 - `mksCmdRunnerGetDefaultHostname`
 - `mksCmdRunnerGetDefaultUsername`
 - `mksCmdRunnerGetDefaultPassword`
 - `mksCmdRunnerGetDefaultImpersonationUser`
 - `mksCmdRunnerSetDefaultHostname`
 - `mksCmdRunnerSetDefaultUsername`
 - `mksCmdRunnerSetDefaultPassword`
 - `mksCmdRunnerSetDefaultImpersonationUser`
 - `mksCreateIntegrationPoint`
 - `mksAPIInitialize`
 - `mksCreateSession`
 - `mksIntegrationPointGetHostname`
 - `mksSessionGetUsername`
 - `mksSessionGetPassword`

User Item Data Type

The user item data type's ID always contains the user ID field (the user's login ID), and may also have an additional `FullName` field, depending on the support for user full names from the MKS Integrity Server. For more information on full name support, see the *MKS Integrity Server 2006 Administration Guide*.

Date and Datetime Data Types

Date and time fields display the time as midnight, based on the local timezone of the client that is running the API.

If you are running on a UNIX platform, you can set the timezone either globally or use the `$TZ` environment variable to set it locally.

Integrity Manager Commands

about

This view command can be used within the API to return product information.

Work Item

The application that product information is returned for.

columnsets/viewcolumnset

This view command can be used within the API to return a summary of MKS Integrity column sets

Work Item

The column set that information is returned for.

connect

This action command can be used within the API to establish a connection to an Integrity Server.

copyissue

This creation command can be used within the API to create a new MKS Integrity issue by copying the common fields of an existing issue. The list of issue types available in your database varies depending on the types created by your administrator and what issue type your workflow permits you to submit.

Command Result

Specifies the issue ID that was created.

cpattributes/viewcpattribute

This view command can be used within the API to return a summary of MKS Integrity change package attributes for a change package type.

For information on these commands and their options, see the *MKS Integrity Server 2006 Administration Guide*.

Work item

The change package attribute that information is returned for.

Work Item Context

The change package type.

Work Item Result

The fields that show for each work item are dependent on the change package type and the attribute data type, for example, a `Logical` field will not have any `Strings` values.

The following table lists the fields added by the API for change package attributes. For information on all available fields, see the *MKS Integrity Server 2006 Administration Guide*.

Field	Data Type	Description
Name	String	Real name of the change package attribute
DisplayName	String	Name displayed to users
Description	String	Detailed description of the attribute
Type	String	Data type of the change package attribute
Position	Integer	Position of the attribute in the change package attribute list
IsReadOnly	Boolean	Specifies if the value of the attribute is read-only
IsMandatory	Boolean	Specifies if a value for the attribute is mandatory
MaxLength	Integer	Maximum length of the change package attribute value
DecimalPlaces	Integer	Number of decimal places allowed in change package attribute value
DisplayFormat	String	Format to display attribute value in
ID	Integer	Identification number of the change package attribute
Strings	List of strings	A list of possible values

[cpentryattributes/viewcpentryattribute](#)

This view command can be used within the API to return a summary of MKS Integrity change package entry attributes for a change package type.

For information on these commands and their options, see the *MKS Integrity Server 2006 Administration Guide*.

Work item

The change package entry attribute that information is returned for.

Work Item Context

The change package type.

Work Item Result

The fields that show for each work item are dependent on the change package type and the entry attribute data type, for example, a `Logical` field will not have any `Strings` values.

The following table lists the fields added by the API for change package entry attributes. For information on all available fields, see the *MKS Integrity Server 2006 Administration Guide*.

Field	Data Type	Description
Name	String	Real name of the change package entry attribute
DisplayName	String	Name displayed to users
Description	String	Detailed description of the entry attribute
Type	String	Data type of the change package entry attribute
Position	Integer	Position of the attribute in the change package entry attribute list
IsReadOnly	Boolean	Specifies if the value of the entry attribute is read-only
IsMandatory	Boolean	Specifies if a value for the entry attribute is mandatory
MaxLength	Integer	Maximum length of the change package entry attribute value
DecimalPlaces	Integer	Number of decimal places allowed in entry attribute value
DisplayFormat	String	Format to display entry attribute value in
ID	Integer	Identification number of the change package entry attribute
Strings	List of strings	A list of possible values

cps/viewcp

This view command can be used within the API to return a summary of MKS Integrity change packages.

Work Item

The change package that information is returned for.

Work Item Fields

In addition to the fields as specified by the `--attributes` and `--entryAttributes` options, the API adds the following field for organizing the list of entry items for the change package being viewed. For information on all available fields, see the *MKS Integrity 2006 CLI Reference Guide*.

Field	Data Type	Description
MKSentries	List of entry items	List of entries in change package. Only available for <code>viewcp</code> command. For information on the change package entry fields, see the <i>MKS Integrity 2006 CLI Reference Guide</i> .

cptypes/viewcptype

This view command can be used within the API to return a summary of MKS Integrity change package types.

For information on these commands and their options, see the *MKS Integrity Server 2006 Administration Guide*.

Work item

The change package type that information is returned for.

Work Item Result

The following table lists the fields added by the API for change package types. For information on all available fields, see the *MKS Integrity Server 2006 Administration Guide*.

Field	Data Type	Description
ID	Integer	Identification number of type
Name	String	Real name of the change package type

Field	Data Type	Description
DisplayName	String	Name displayed to users
Position	Integer	Order number for type
Description	String	Detailed description of the change package type
Attributes	Change package attribute item	List of change package attributes associated with the change package type. For information on the change package attribute fields, see the <i>MKS Integrity Server 2006 Administration Guide</i> .
EntryAttributes	Change package attribute items	List of change package entry attributes associated with the change package type. For information on the change package entry attribute fields, see the <i>MKS Integrity Server 2006 Administration Guide</i> .
EntryKey	Change package entry attribute items	List of change package entry attributes to be used as a key (to uniquely identify change package entries). For information on the change package entry attribute fields, see the <i>MKS Integrity Server 2006 Administration Guide</i> .
PermittedGroups	List of groups	Groups allowed to use this type.
PermittedAdministrators	List of users or groups	Users or groups allowed to perform administration operations on type.

createcp

This creation command can be used within the API to create an MKS Integrity change package.

For information on this command and its options, see the *MKS Integrity Server 2006 Administration Guide*.

Command Result

Specifies the change package ID that was created.

createcpattribute

This creation command can be used within the API to create an MKS Integrity change package attribute for a change package type.

For information on this command and its options, see the *MKS Integrity Server 2006 Administration Guide*.

Command Result

Specifies the change package attribute that was created.

Result Context

The change package type.

createcpenentry

This creation command can be used within the API to create an MKS Integrity change package entry.

For information on this command and its options, see the *MKS Integrity Server 2006 Administration Guide*.

Command Result

Specifies the change package ID that the entry was created for.

createcpenentryattribute

This creation command can be used within the API to create an MKS Integrity change package entry attribute for a change package type.

For information on this command and its options, see the *MKS Integrity Server 2006 Administration Guide*.

Command Result

Specifies the change package entry attribute that was created.

Result Context

The change package type.

createcptype

This creation command can be used within the API to create a new MKS Integrity change package type. Common change package attributes will automatically be created.

For information on this commands and its options, see the *MKS Integrity Server 2006 Administration Guide*.

Command Result

Specifies the change package type created.

createissue

This creation command can be used within the API to create an MKS Integrity issue. The list of issue types available in your database varies depending on the types created by your administrator and what issue type your workflow permits you to submit.

Command Result

Specifies the issue ID number that was created.

deletecp

This action command can be used within the API to delete an MKS Integrity change package.

Command Result

Specifies the change package ID that was deleted.

deletecpattribute

This action command can be used within the API to delete one or more MKS Integrity change package attributes for a change package type.

For information on this command and its options, see the *MKS Integrity Server 2006 Administration Guide*.

Command Result

Specifies the change package attribute that was deleted.

Result Context

The change package type.

deletecpentry

This action command can be used within the API to delete an MKS Integrity change package entry.

For information on this command and its options, see the *MKS Integrity Server 2006 Administration Guide*.

Command Result

Specifies the change package ID that the entry was deleted from.

deletecpendryattribute

This action command can be used within the API to delete one or more MKS Integrity change package entry attributes for a change package type.

For information on this command and its options, see the *MKS Integrity Server 2006 Administration Guide*.

Command Result

Specifies the change package entry attribute that was deleted.

Result Context

The change package type.

deletecptype

This action command can be used within the API to delete one or more MKS Integrity change package types.

For information on this command and its options, see the *MKS Integrity Server 2006 Administration Guide*.

Command Result

Specifies the change package type deleted.

disconnect

This action command can be used within the API to disconnect from the Integrity Server.

Command Result

Specifies the Integrity Server that was disconnected.

editcpattribute

This action command can be used within the API to make changes to an MKS Integrity change package attribute.

For information on this command and its options, see the *MKS Integrity Server 2006 Administration Guide*.

Work Item

The change package attribute that is to be updated.

Work Item Context

The change package type.

Work Item Result

Specifies the change package attribute that was updated.

editcpentry

This action command can be used within the API to make changes to an MKS Integrity change package entry.

For information on this command and its options, see the *MKS Integrity Server 2006 Administration Guide*.

Work Item

The change package ID that is to be updated.

Work Item Result

Specifies the change package ID that was updated.

editcpentryattribute

This action command can be used within the API to make changes to an MKS Integrity change package entry attribute.

For information on this command and its options, see the *MKS Integrity Server 2006 Administration Guide*.

Work Item

The change package entry attribute that is to be updated.

Work Item Context

The change package type.

Work Item Result

Specifies the change package entry attribute that was updated.

editcptype

This action command can be used within the API to make changes to an MKS Integrity change package type.

For information on this command and its options, see the *MKS Integrity Server 2006 Administration Guide*.

Work Item

The change package type that is to be updated.

Work Item Result

Specifies the change package type that was updated.

editissue

This action command can be used within the API to make changes to an MKS Integrity issue.

NOTE If there is a problem, this command does not retry the edit, so all mandatory fields must be specified as required when the command is executed.

Work Item

The issue that is to be updated.

Work Item Result

Specifies the issue that was updated.

exit

This action command can be used within the API to exit the current MKS Integrity client session.

extractattachments

This action command can be used within the API to save one or more attachments from an MKS Integrity issue.

Work Item

The filename of the attachment.

Work Item Context

The MKS Integrity issue that the attachment belongs to.

fields/viewfield

This view command can be used within the API to return a list of fields. By default, all fields are selected to be returned.

Work Item

The field that information is returned for.

Work Item Fields

The fields that show for each work item depend on the field type, for example, a `String` field will not have `Min` or `Max` values. For more information on the data types for fields, see the *MKS Integrity Server 2006 Administration Guide*.

groups/viewgroup

This view command can be used within the API to return a list of groups. By default, all groups are selected to be returned.

Work Item

The group that information is returned for.

importissue

This action command can be used within the API to import an issue into the MKS Integrity database, creating a new MKS Integrity issue to represent it.

Command Result

Specifies the issue ID that was imported into MKS Integrity.

issues/viewissue

This view command can be used within the API to return a list of MKS Integrity issues.

Work Item

The issue that information is returned for.

Work Item Fields

The following table lists the fields added by the API for issues. For information on all available fields, see the *MKS Integrity 2006 CLI Reference Guide*.

Field	Data Type	Description
MKSIssueAttachments	List of attachment items	Attachments associated with issue. Only available for viewissue command. For more information on the issue attachment fields, see the <i>MKS Integrity 2006 CLI Reference Guide</i> .
MKSIssueHistory	List of issue change items	Historical changes made to issue. Only available for viewissue command. For more information on the issue history fields, see the <i>MKS Integrity 2006 CLI Reference Guide</i>
MKSIssueRelationship	List of related issues	Issues related to the issue. Only available for viewissue command. For more information on the issue relationship fields, see the <i>MKS Integrity 2006 CLI Reference Guide</i>
MKSIssueTimeEntries	List of time entries	Time entries associated with the issue. Only available for viewissue command. For more information on the issue time entries fields, see the <i>MKS Integrity 2006 CLI Reference Guide</i>

projects/viewproject

This view command can be used within the API to return a list of projects. By default, all projects are selected to be returned.

Work Item

The project that information is returned for.

Work Item Fields

The following table lists the fields added by the API for projects. For information on all available fields, see the *MKS Integrity 2006 CLI Reference Guide*.

Field	Data Type	Description
Name	String	Name of project
Description	String	Description of project
ID	Integer	Identification number of project
Parent	String	Parent project
InheritPermittedGroups	Boolean	Specifies if project inherits permitted groups from its parent. If this field is set to <code>true</code> , then PermittedGroups field will be empty. To find permitted groups, you need to look at parent (and possibly its parent).
PermittedGroups	List of groups	Groups of users who can work with project. Item ID is group name.
PermittedAdministrators	List of users or groups	Users or groups allowed to perform administration operations on project.

queries/viewquery

This view command can be used within the API to return a list of MKS Integrity queries. By default, all queries that are currently shared with you are returned.

Work Item

The query that information is returned for.

Work Item Context

The name of the user who created the query.

servers

You can use this view command within the API to return the current connections to an MKS Integrity Server.

Work Item

The server connection that information is returned for.

states/viewstate

You can use this view command within the API to return a list of issue states in MKS Integrity. By default, all states are listed.

Work Item

The issue state that information is returned for.

types/viewtype

You can use this view command within the API to return a list of MKS Integrity issue types. By default, all types are returned.

Work Item

The issue type that information is returned for.

Work Item Fields

The following table lists the fields added by the API for issue types. For information on all available fields, see the *MKS Integrity 2006 CLI Reference Guide*.

Field	Data Type	Description
MandatoryFields	List of mandatory field items	Mandatory fields for issue type.
VisibleFields	List of field items	Visible fields for issue type.
StateTransitions	List of state transition items	State transitions for issue type.
FieldRelationships	List of field items	Related fields for issue type.
PermittedAdministrators	List of users or groups	Users or groups allowed to perform administration operations on type.

users/viewuser

You can use this view command within the API to display a list of users. By default, all users are displayed.

Work Item

The user that information is returned for.

viewcolumnset

This view command can be used within the API to return detailed information on a selection of column sets.

For more information, see “columnsets/viewcolumnset” on page 64.

viewcp

This view command can be used within the API to return detailed information on a selection of change packages.

For more information, see “cps/viewcp” on page 67.

viewcpatrick

This view command can be used within the API to return detailed information on a selection of change package attributes.

For more information, see “cpattributes/viewcpatrick” on page 64.

viewcpenrypatrick

This view command can be used within the API to return detailed information on a selection of change package entry attributes.

For more information, see “cpenryattributes/viewcpenrypatrick” on page 65.

viewcptype

This view command can be used within the API to return detailed information on a selection of change package type.

For more information, see “cptypes/viewcptype” on page 67.

viewfield

This view command can be used within the API to return detailed information on a selection of MKS Integrity fields.

For more information, see “fields/viewfield” on page 74.

viewgroup

This view command can be used within the API to return detailed information on a selection of MKS Integrity groups.

For more information, see “groups/viewgroup” on page 74.

viewissue

This view command can be used within the API to return detailed information on a selection of MKS Integrity issues.

For more information, see “issues/viewissue” on page 74.

viewproject

This view command can be used within the API to return detailed information on a selection of MKS Integrity projects.

For more information, see “projects/viewproject” on page 75.

viewquery

This view command can be used within the API to return detailed information on a selection of MKS Integrity queries.

For more information, see “queries/viewquery” on page 76.

viewstate

This view command can be used within the API to return detailed information on a selection of MKS Integrity issue states.

For more information, see “states/viewstate” on page 77.

viewtype

This view command can be used within the API to return detailed information on a selection of MKS Integrity issue types.

For more information, see “types/viewtype” on page 77.

viewuser

This view command can be used within the API to return detailed information on a selection of MKS Integrity users.

For more information, see “users/viewuser” on page 77.

Source Integrity Commands

about

This view command can be used within the API to return product information.

Work Item

The application that product information is returned for.

add

This action command can be used within the API to add one or more nonmember files to an MKS Source Sandbox.

NOTE Only an Integrity Client 2006 connecting to an Integrity Server 2006 properly displays status messages for all members. If you are using a different version of the Integrity Client or Integrity Server, display status messages are not guaranteed.

Work Item

The file that is to be added to the Sandbox.

Work Item Context

The Sandbox that the file was added to.

Work Item Result

Specifies the file that was added to the Sandbox. The result can be one of the following:

- member
- destined member (for deferred additions)

Work Item Result Context

The project the file is being added to (none for deferred additions).

addsubproject

This action command can be used within the API to add one or more subprojects to a MKS Source project.

Work Item

The location of the subproject to be added.

Work Item Context

The project the subproject was added to.

Work Item Result

Specifies the path of the subproject that was added.

Work Item Result Context

The name of the project the subproject was added to.

archiveinfo

This view command can be used within the API to return information on an archive.

Work Item

The member that archive information is returned for.

Work Item Context

The project path.

Work Item Fields

The following table lists the fields added by the API for archives. For information on all available fields, see the *MKS Source 2006 CLI Reference Guide*.

Field	Data Type	Description
MemberName	String	Path and name of member archive
SandboxName	String	Path and name of member's Sandbox
ProjectName	String	Path and name of the member's project
ArchiveName	String	Path and name of archive
Archivetype	String	Type of data stored in archive. This can be one of following: Text, Binary.

Field	Data Type	Description
StrictLocking	Boolean	Specifies if there is a strict locking policy in effect for archive
Compressed	Boolean	Specifies if archive is compressed
StorebyReference	Boolean	Specifies if each revision is saved to a separate file
ArchiveDescription	String	Description of archive
Labels	List of label items	Revision labels in archive
Locks	List of lock items	Locks in archive
DeployType	String	The deploy type for the archive if the member belongs to a deploy project. This field is only visible if you are using MKS Deploy. For more information, see the <i>MKS Deploy 2006 Administration Guide</i> .

ci

This action command can be used within the API to check in members of a Sandbox.

Work Item

The member that is checked in.

Work Item Context

The project the file is being checked into.

Work Item Result

Specifies the checked in revision.

The following table lists the potential field for the command result.

Field	Data Type	Description
MemberRev	Revision item	Revision number of member after checkin. Item ID is revision number.

closecp

This action command can be used within the API to close change packages.

Work Item

The change package ID of the change package to be closed.

Work Item Result

Specifies that the change package was closed.

co

This action command can be used within the API to check out members into working files in a Sandbox.

Work Item

The member that is to be checked out.

Work Item Context

The project the file is being checked out of.

Work Item Result

Specifies the revision of the member that was checked out.

The following table lists the potential field for the command result.

Field	Data Type	Description
IsLocked	Boolean	Specifies if member is locked

connect

This action command can be used within the API to establish a connection to an Integrity Server.

configuresubproject

This action command can be used within the API to configure a subproject in a MKS Source project.

Work Item

The location of the subproject to be configured.

Work Item Context

The project location the subproject belongs to.

Work Item Result

Specifies the path of the subproject that was configured.

Work Item Result Context

The name of the project the subproject belongs to.

createtcp

Use this creation command within the API to create a change package for an MKS Integrity issue.

Command Result

Specifies the change package ID of the change package that was created.

createproject

Use this creation command within the API to create a new, empty project on the Integrity Server in a specified directory.

Command Result

Specifies the project that was created.

createsandbox

Use this creation command within the API to create a new Sandbox on your local machine.

Command Result

Specifies the Sandbox that was created.

Result Context

The project the Sandbox was created from.

createsubproject

Use this creation command within the API to create a new subproject for a MKS Source project.

Command Result

Specifies the subproject that was created.

Result Context

The project the subproject was created for.

disconnect

This action command can be used within the API to disconnect from the Integrity Server.

Command Result

Specifies the Integrity Server that was disconnected.

drop

Use this action command within the API to drop one or more member files or subprojects from a project.

NOTE Only an MKS Integrity Client 2006 connecting to an MKS Integrity Server 2006 properly displays status messages for all members. If you are using a different version of the Integrity Client or Integrity Server, display status messages are not guaranteed.

Work Item

The work item can be one of the following:

- member that was dropped
- subproject that was dropped

The type of work item can be determined by the model type.

Work Item Context

The context can be one of the following:

- for member work items, the full path of the (sub)project for the member
- for former members, the full path of the Sandbox for the member

exit

This action command can be used within the API to exit the current MKS Source client session.

gui

This action command can be used within the API to start the MKS Source graphical user interface (GUI).

lock

This action command can be used within the API to lock project members.

Work Item

The member that was locked.

Work Item Context

The project name for the file being locked.

Work Item Result

Identifies the locked revision.

The following table lists the potential field for the command result.

Field	Data Type	Description
BranchCreated	Boolean	Specifies if a branch was created for member that was locked

locks

This view command can be used within the API to return information on a user’s locks.

Work Item

The locked revision.

Work Item Context

The archive that the locked revision belongs to.

projectadd

This action command can be used within the API to add an existing project to MKS Source.

Work Item

The project that is to be added.

Work Item Result

Specifies the path of the project that was added.

projectci

This action command can be used within the API to check in members of a project.

Work Item

The member that is checked in.

Work Item Context

The project the file is being checked into.

Work Item Result

Specifies the checked in revision.

The following table lists the potential field for the command result.

Field	Data Type	Description
MemberRev	Revision item	Revision number of member after checkin. Item ID is revision number.

projectco

This action command can be used within the API to check out members into working files.

Work Item

The member that is to be checked out.

Work Item Context

The project the file is being checked out of.

Work Item Result

Specifies the revision of the member that was checked out.

The following table lists the potential field for the command result.

Field	Data Type	Description
IsLocked	Boolean	Specifies if member is locked

projectinfo

You can use this view command within the API to return information for a project.

Work Item

The project that information is being returned for.

Work Item Fields

The following table lists the fields added by the API for projects. For information on all available fields, see the *MKS Source 2006 CLI Reference Guide*.

Field	Data Type	Description
ProjectName	String	Name of project
Server	String	Server that project resides on
ServerPort	String	Port of server that project resides on
DevelopmentPath	String	Development path name
FullConfigSyntax	String	Configuration path of project
Revision	Item	Revision number of project. Item ID is revision number.
LastCheckpoint	Date	Date and time project was last checkpointed
NumMembers	Integer	Number of members in project
NumSubprojects	Integer	Number of subprojects in project
Description	String	Detailed description of project
ProjectAnnotation	String	Information describing the use of the project in MKS Deploy. You only see this field if you are licensed to use MKS Deploy. For more information, see the <i>MKS Deploy 2006 Administration Guide</i> .
ProjectAttributes	List of attribute items	Attributes of project

Field	Data Type	Description
DevelopmentPaths	List of development path items	Development path of project
ACLName	String	Name of access control list (ACL) for project
ACLExists	Boolean	Specifies if project ACL exists
ActualACL	String	Name of real ACL

rename

You can use this action command within the API to rename an existing project member.

Work Item

The member that is to be renamed. This is the “old” name of the member, prior to the rename operation.

Work Item Context

The project path of the file.

Work Item Result

Specifies the new name of the renamed member.

Work Item Result Context

The project path of the file.

Result Fields

The following table lists the potential field for the work item result.

Field	Data Type	Description
RenameWorkingFile	Boolean	Specifies if working file for member was also renamed

resync

You can use this action command within the API to update a Sandbox with the member revision.

Work Item

The work item can be one of the following:

- member that was resynced
- subsandbox that was updated
- former member that was resynced

The type of work item can be determined by the model type.

Work Item Context

The work item context can be one of the following:

- for members, the relative name of the subproject it is in or, if it is in the top-level Sandbox, the full path to the Sandbox
- for former members, the absolute path in the Sandbox for the member

Work Item Result

Specifies the revision that was checked out (if needed).

revert

You can use this action command within the API to overwrite a Sandbox file with a fresh copy of the working file, discarding changes.

Work Item

The work item can be one of the following:

- member that was reverted
- former member that was reverted

Work Item Context

The work item context can be one of the following:

- for members, the relative subproject the member is in, or if the member is in the top-level project, the full path name to the project
- for former members, the absolute path in the Sandbox for the member

Work Item Result

Specifies the working file that overwrote the Sandbox file.

The following table lists the potential fields for the command result.

Field	Data Type	Description
DeletedWorkingFile	Boolean	Specifies if working file was removed
Unlocked	Boolean	Specifies if working revision was unlocked
RemovedAdd	Boolean	Specifies if add operations were removed
RemovedCheckin	Boolean	Specifies if check in operations were removed
RemovedDrop	Boolean	Specifies if drop operations were removed
RemovedImport	Boolean	Specifies if import operations were removed
RemovedRename	Boolean	Specifies if rename operations were removed
RemovedUpdateRevision	Boolean	Specifies if update revision operations were removed
RemovedMoveMember	Boolean	Specifies if move operations were removed

sandboxinfo

Use this view command within the API to return information about a Sandbox.

Work Item

The Sandbox that information is returned for.

Work Item Context

The project the Sandbox references.

Work Item Fields

The following table lists the fields added by the API for Sandboxes. For information on all available fields, see the *MKS Source 2006 CLI Reference Guide*.

Field	Data Type	Description
SandboxName	String	Name of Sandbox
ProjectName	String	Name of project
Server	String	Server that project resides on
ServerPort	String	Port of server that project resides on
DevelopmentPath	Development path item	Development path of Sandbox. Item ID is development path label.

Field	Data Type	Description
FullConfigSyntax	String	Configuration path of project
Revision	Item	Revision number of project. Item ID is revision number.
LastCheckpoint	Date	Date and time project was last checkpointed
NumMembers	Integer	Number of members in project
NumSubprojects	Integer	Number of subprojects in project
Description	String	Detailed description of project
ProjectAnnotation	String	Information describing the use of the project in MKS Deploy. You only see this field if you are licensed to use MKS Deploy. For more information, see the <i>MKS Deploy 2006 Administration Guide</i> .
Sparse	Boolean	Specifies if Sandbox is sparse
Shared	Boolean	Specifies if Sandbox is shared
ProjectAttributes	List of attribute items	Attributes of project
SandboxAttributes	List of attribute items	Attributes of Sandbox
LineTerminator	String	Line terminator used for Sandbox

unlock

You can use this action command within the API to unlock a member.

Work Item

The member that was unlocked.

Work Item Context

The project name.

Work Item Result

Specifies the revision that was unlocked.

The following table lists the potential field for the command result.

Field	Data Type	Description
BrokenLockHolder	User item	User whose lock was broken/removed.

viewhistory

You can use this view command within the API to return member history information.

Work Item

The member that the history information is returned for.

Work Item Context

The project the member belongs to.

Work Item Fields

The following table lists the fields added by the API for member histories. For information on all available fields, see the *MKS Source 2006 CLI Reference Guide*.

Field	Data Type	Description
Revisions	List of revision items	Member revisions. For information on revision fields, see the <i>MKS Source 2006 CLI Reference Guide</i> .

viewproject

Use this view command within the API to return the contents of a project and some information on its members.

Work Item

The work item can be one of the following:

- member that information is returned for
- subproject that information is returned for

Work Item Context

The context for a member work item is the project.

viewsandbox

Use this view command within the API to return the contents of a Sandbox.

Work Item

The work item can be one of the following:

- Sandbox member that information is returned for
- former member that information is returned for
- subsandbox that information is returned for
- former subsandbox that information is returned for

Work Item Context

The work item context can be one of the following:

- for a member/former member, the parent Sandbox
- for a Sandbox/former Sandbox, no context

Administering the MKS API

Setting up Integrity Suite for the MKS API

This chapter provides information on how to set up and administrate the MKS API in MKS Integrity.

It contains information on the following:

- “Setting Up Permissions” on page 96
- “Setting Up Policies” on page 96
- “Setting Up Impersonations” on page 97
- “Backward Compatibility” on page 98
- “Patch Management” on page 99
- “Troubleshooting” on page 100
- “Supported Configurations” on page 101

Setting Up Permissions

To be able to run certain commands, all users of the integration must be set up to have permission to view MKS Integrity administrative information. The ability to view MKS Integrity administrative information is controlled through Access Control Lists (ACLs). For more information, see the *MKS Integrity Server 2006 Administration Guide*.

Setting Up Policies

There are two policies that control access to the Integrity Client and the Integrity Server:

- connection policy
- authentication policy

These policies are specified in the `IntegrityClient.rc` file on the Integrity Client (`.IntegrityClient.rc` for UNIX), and in the `config/IntegrityClientSite.rc` file on the Integrity Server. These files are located in your home directory on the client or install directory of the server.

The connection policy is specified by the `daemon.connectionPolicy`. The authentication policy is specified by the `daemon.authenticationPolicy`. The settings for these policies depend on your integration point. MKS has set defaults for both the client and the server integration points.

Client Integration Point Policy Defaults

The following defaults are set for Windows clients.

- `mks.ic.common.policy.ICLocalConnectionPolicy`
Allows connections from `localhost` only.
- `mks.ic.common.policy.ICNoAuthenticationPolicy`
Allows everything to connect.

The following defaults are set for UNIX clients.

- `mks.ic.common.policy.ICAllowAllConnectionPolicy`
Allows everything to connect.
- `mks.ic.common.policy.ICLocalAuthenticationPolicy`
Validates the user through a local system file.

Server Integration Point Policy Defaults

The following defaults are set for servers.

- `mks.ic.common.policy.IAllowSpecificConnectionPolicy`
Allows a specific set of IP addresses to connect. You must specify the IP addresses in comma-delimited format for the `daemon.validConnectionList` property.
- `mks.ic.common.policy.IAuthenticateSessionPolicy`
Validates the standard HTTP authorization credentials against an external URL. Each session can have a different set of credentials associated with it. Initially, a session calls the URL to make sure the credentials are valid. On subsequent requests for that session, the credentials passed in on the request are compared to the previously stored validated credentials.

By default, the MKS authentication servlet (`http://host:port/Authenticate`) is used to perform the validation against the configured realms.

You can also use `IAllowAllConnectionPolicy` as the connection policy for a server integration point. If you use this policy you should specify the users who are allowed to connect in comma-delimited format for the `daemon.validUsersList` property. If you do not set this property, all authenticated users are allowed to connect.

Client as Server Integration Point Policies

No policy defaults are set for a client as server integration point. If you are using this scenario, the recommended defaults are the same as for a server integration scenario.

Setting Up Impersonations

The ability to impersonate and the users who can be impersonated are controlled through ACLs. For more information on ACLs, see the *MKS Integrity Server 2006 Administration Guide*. Currently, the impersonation ACLs can only be set up through the Authorization Administration Web interface or the CLI; they cannot be configured through the Integrity Server Administration Client.

For information on configuring an ACL using the Web interface, see the *MKS Integrity Server 2006 Administration Guide*. For information on working with the Authorization Administration command line interface, see the *MKS Integrity Server 2006 Administration CLI Reference Guide*.

The format of the impersonation ACL can be either:

```
mks:impersonate:user:<user name>
```

or

```
mks:impersonate:group:<group name>
```

The ACL name identifies the specific user or user group that can be impersonated. The principal of the ACL is the specific user or group who is either allowed or denied permission to impersonate the user group specified in the ACL name.

For example, with an ACL of `mks:impersonate:user:bill`, if the user group “Manager” is listed as a principal and is allowed permission, anyone in the Manager user group can impersonate Bill.

The following rules apply to impersonation ACLs:

- The impersonation permission is not available if there are no ACLs.
- Individual user ACLs take precedence over group ACLs.
- If a requesting user belongs to multiple groups, some of which are allowed to impersonate a particular user or group, and some of which are explicitly denied the impersonation of the user or group, the requesting user is *not* allowed to perform the impersonation.

For more information on using impersonation, see “Using Impersonation” on page 45.

Backward Compatibility

The MKS API is designed to minimize the frequency and impact of integration changes due to MKS product upgrades and patches. Backward compatibility features allow newer clients and servers to continue to work with older versions of the MKS API. There are two aspects to backward compatibility:

- Java and C API compatibility
- command compatibility

Java and C API Compatibility

The functions and behavior of the API only change when the MKS API is either enhanced or patched. These changes will be clearly identified with the new release and must be accommodated in any code that is being upgraded to use the new API version. To avoid unnecessary changes to your integration, only use a newer version of the API when it contains patches or features you need.

Command Compatibility

Command behavior is determined by the integration point. So upgrading either a client or server has the potential to alter a command run through the API. Changes that could affect API commands include new commands, new command options, or changed command defaults, functions, or exceptions.

Where possible, the integration point recognizes an older API version and attempts to accommodate it. This means that if an integration using the API from Integrity Manager 2005 is connected to an integration point running MKS Integrity 2006, the integration point would attempt to specify the command and organize the response as if it was in Integrity Manager 2005. For example, if a new option is added to a command, the integration point recognizes that the option did not exist when the integration was written. So the option default is set to the value that mimics the prior release behavior.

Not all future product changes can be accommodated. These exceptions will be noted in future versions of this manual and may require integration changes.

Older API versions can be used to run new commands or new command options. Since all commands are interpreted by the integration point, if the integration point has been upgraded or patched to include new commands or command options, they can be run through an older API version that can talk to that integration point.

Patch Management

When executing commands on an Integrity Client, the patch manager is disabled. No patches are checked/installed as result of API connections to a client. Although the command **updateclient** can be run through the API, it is not recommended since it requires interaction and often requires a restart of the client.

Troubleshooting

If you encounter a problem with the MKS API, contact MKS Customer Care as outlined in “Getting Help” on page 6.

You may find the answer to your problem on the MKS Developer Zone page of the MKS Web site (www.mks.com/devzone). There you will find knowledge base articles with an integration focus, a user forum dedicated to the discussion of integration issues, sample integration applications, and the latest information on MKS integrations in general.

If you need to contact a Technical Support Representative, you should gather information using the methods described below.

Using the API Viewer

By using the API Viewer to re-create your problem and sending the result to your Technical Support Representative, you help isolate API problems from integration code problems.

API Version Number

It is important that your Technical Support Representative know which version of the MKS API you are working with. The API version may be different than the MKS Integrity version.

To find the version number of the Java API

You can find the version of the API in the manifest of the `mksapi.jar` file. The API version is identified as the Implementation-Version and uses the following format:

<MKS Integrity version> <patch version> <build version>

This information is also available in the `mksapi.log` file.

To find the version number of the C API

You can find the version of the C API in the `mksVersion.h` file found in the Integrity client `lib\include` directory. The API version is identified as the Implementation-Version and uses the following format:

<MKS Integrity version> <patch version> <build version>

Using API Logging

By enabling API logging, you can capture information that helps your Technical Support Representative solve your problem.

To enable logging for the Java API

Logging must be enabled prior to getting the first instance of `IntegrationPointFactory`. To enable API logging, use the `MKSLogger(logfile)` constructor. All log data is sent to the file specified by `logfile` or `system.out` if null is passed in.

For the Java API, different logging methods can be defined. For more information on the methods available see the MKS Java API documentation accessed through the Integrity Server Web page.

To enable logging for the C API

Calling the `mksAPIInitialize(char* logfile)` function initializes logging. Log data is sent to the log file specified. If the parameter is `NULL`, then the log data is sent to `mksapi.log` in the current working directory. Once the `mksAPIInitialize` function is called, the logging behavior cannot be dynamically changed.

For the C API, different logging mechanisms can be defined. For information on the mechanisms available see the MKS ANSI C API documentation accessed through the Integrity Server Web page.

Supported Configurations

The MKS API supports the following configurations.

- **Supported configurations for the Java API**

The Java API is certified for Java Runtime Environment 1.5, but only uses functions that are available in JRE 1.2. Although no functional tests have been performed by MKS on JRE 1.2, the API should still work correctly for that version.

- **Supported configurations for C**

The C version of the API is implemented using ANSI C. You can use the C API with any ANSI C compiler on supported platforms. See the *MKS Integrity Server 2006 Installation Guide* for supported platforms.

NOTE You cannot use the C API with Integrity products prior to version 2005.

Index

A

- about command 64
- access method 11
- action commands 27
- add command 80, 81
- administrating the MKS API 95–101
- API model 10
- API viewer 32
 - running for C API 33
 - running for Java API 32
 - using for troubleshooting 100
- application error codes for C API 40
- application errors for C API 39
- archiveinfo command 81
- assumptions for this guide 5

B

- backward compatibility 98
- best practices 47
 - converting Source Integrity scripts 54
 - dedicated sessions 52
 - including a copy of the API 48
 - managing users 48
 - memory management tips 53
 - using facade patterns 54

C

- C API
 - application error codes 40
 - application errors 39
 - creating a session for 23
 - creating an integration point for 22
 - creating command runner for 24
 - creating program 56
 - documentation for C functions 18
 - errors 39
 - internal error codes 41
 - logging errors for 101
 - running the viewer utility for 33

- specifying a command for 25
- supported configurations for 101
- unicode support 63
- C functions documentation 18
- caching sessions 22
- certified commands 59–94
 - list of 61
- ci command 82
- client as server integration
 - description 13
 - policies for 97
- client integration
 - description 12
 - policy defaults 96
- co command 82, 83
- command 60
 - about 64
 - action 27
 - add 80, 81
 - archiveinfo 81
 - certified 59–94
 - certified for MKS Integrity 61
 - certified for MKS Source 61
 - ci 82
 - co 82, 83
 - configure subproject 83
 - connect 64, 83
 - copyissue 64
 - cpattribute 64
 - cpentryattributes 65
 - cptypes 67
 - createcp 68, 84
 - createcpattribute 68
 - createcpentry 69
 - createcpentryattribute 69
 - createcptype 69
 - createissue 70
 - createproject 84
 - createsandbox 84
 - creation type 26
 - default behavior 25
 - deletecp 70
 - deletecpattribute 70

- deletectpentry 70
- deletectpentryattribute 71
- deletectpentrytype 71
- disconnect 71, 85
- drop 85
- editcpattribute 71
- editcpentry 72
- editcpentryattribute 72
- editcpentrytype 73
- editissue 73
- executing 26
- exit 73, 85
- extractattachments 73
- fields 74
- groups 74
- gui 86
- importissue 74
- issues 74
- list of certified commands 61
- lock 86
- locks 86
- projectadd 86
- projectci 87
- projectco 87
- projectinfo 88
- projects 75
- queries 76
- rename 89
- response summary 31
- resync 89
- revert 90
- running 24
- sandboxinfo 91
- servers 76
- specifying 24
 - for C API 25
 - for Java API 24
- states 77
- types 77
- uncertified 47
- unlock 92
- users 77
- using the graphical user interface 43
- variations 42
- view 28
- viewcpattribute 64
- viewcpentryattributes 65
- viewcptypes 67
- viewfield 74
- viewgroup 74
- viewhistory 93
- viewissue 74

- viewproject 75, 93
- viewquery 76
- viewsandbox 94
- viewstate 77
- viewtype 77
- viewuser 77
- command exceptions for Java API 36
- command runner
 - creating 24
 - for C API 24
 - for Java API 24
 - description 23
- common session 23
- compatibility, backward 98
- configure subproject command 83
- connect command 64, 83
- connection exceptions
 - for Java API 36
 - for Java API application 38
- contents of guide 2
- copying the API 48
- copyissue command 64
- cpattribute command 64
- cpentryattributes command 65
- cptypes command 67
- createcp command 68, 84
- createcpattribute command 68
- createcpentryattribute command 69
- createcpentrycommand 69
- createcpentrytype command 69
- createissue command 70
- createproject command 84
- createsandbox command 84
- creation type commands 26
- current working directory for file references 46
- Customer Care 6

D

- data type
 - date 63
 - datetime 63
 - standard Java data types 62
 - user item 63
- date data type 63
- datetime data type 63
- deletectp command 70
- deletectpattribute command 70
- deletectpentry command 70
- deletectpentryattributecommand 71
- deletectpentrytype command 71

desktop with server integration 14
 disconnect command 71, 85
 documentation
 related 2, 5
 drop command 85

E

editcptribute command 71
 editcptrentry command 72
 editcptrentryattribute command 72
 editcptrtype command 73
 editissue command 73
 errors for C API 39
 application error codes 40
 application errors 39
 internal error codes 41
 exceptions for Java API 34
 application connection exceptions 38
 application exceptions 37
 catching 35
 command exceptions 36
 connection exceptions 36
 internal exceptions 39
 interrupted exceptions 39
 item exceptions 38
 launch exceptions 39
 permission exceptions 38
 work item exceptions 36
 exit command 73, 85
 extractattachments command 73

F

facade patterns 54
 feedback on documentation 8
 fields command 74
 fields, standard Java data types 62
 file reference
 effect of integration point 46
 specifying current working directory for 46
 using 46
 file system, accessing 46

G

getting help 6
 getting started with the API 1–8
 graphical user interface
 using 43

 using for prompting 43
 using for status reporting 43
 using for view commands 44
 groups command 74
 gui command 86
 guide contents 2

H

help, accessing 4

I

impersonation
 setting up 97
 using 45
 importissue command 74
 information format 60
 integration
 memory management tips 53
 scenarios 11
 steps to follow 15
 integration point 20
 communicating with 20
 controlling access to 20
 creating 21
 for C API 22
 for Java API 21
 description 11
 effect on file references 46
 memory management tips 53
 interim response 44
 internal error codes for C API 41
 internal exceptions for Java API 39
 interrupted exceptions for Java API 39
 issues command 74
 item field, accessing from responses 55

J

Java API
 application connection exceptions 38
 application exceptions 37
 catching exceptions 35
 command exceptions 36
 connection exceptions 36
 creating a session for 23
 creating command runner for 24
 creating integration point for 21
 documentation for public classes 18
 exceptions 34

- internal exceptions 39
- interrupted exceptions 39
- item exceptions for Java API 38
- launch exceptions 39
- logging errors for 101
- permission exceptions 38
- running the viewer utility for 32
- specifying a command for 24
- standard data types 62
- supported configurations for 101
- work item exceptions 36

K

- knowledge assumptions for this guide 5

L

- launch exceptions for Java API 39
- lock command 86
- locks command 86
- logging
 - C API errors 101
 - Java API errors 101
- logging API errors 100

M

- memory management tips 53
 - within the integration 53
 - within the integration point 53
 - within the response 54
- MKS Integrity certified commands 61
- MKS Source certified commands 61
- model for the API 10
- multiple user session 50

P

- patches, managing 99
- permission
 - exceptions for Java API 38
 - setting up 96
- policies
 - client as server integration policies 97
 - client integration policy defaults 96
 - server integration policy defaults 97
 - setting up 96
- program
 - creating for C 56

- project add command 86
- projectci command 87
- projectco command 87
- projectinfo command 88
- projects command 75
- prompting using the graphical user interface 43

Q

- queries command 76

R

- recommended practices 47
- related documentation 2, 5
- rename command 89
- response
 - accessing item fields from 55
 - for action commands 27
 - for creation type commands 26
 - memory management tips 54
 - reading 26
 - retrieving view information from 28
 - retrieving work item results from 28
 - retrieving work items from 27
 - summary 31
 - using interim 44
 - viewing with the API viewer 32
- resync command 89
- revert command 90

S

- sandboxinfo command 91
- script, converting to API programs 54
- server integration
 - description 13
 - policy defaults 97
- servers command 76
- services, professional 7
- session 22
 - caching 22
 - creating 23
 - for C API 23
 - for Java API 23
 - dedicated to a single user 52
 - multiple users 50
 - single user session 48
 - using a common session 23
- standardized interface using facade patterns 54
- states command 77

status reporting using the GUI 43
 support 6
 supported configurations 101

T

troubleshooting 100
 API version number 100
 using API logging 100
 for the C API 101
 for the Java API 101
 using the API viewer 100
 types command 77
 typographical conventions 5

U

uncertified commands 47
 understanding the MKS API 9–15
 unicode support for C API 63
 unlock command 92
 user
 dedicated session for 52
 managing 48
 managing multiple users for a session 50
 single user session 48
 user item data type 63
 users command 77
 using the API 17–56

V

version number of API 100
 view command
 retrieving view information for 28
 using the graphical user interface for 44
 view commands 28
 viewcpattribute command 64
 viewcpentryattributes command 65
 viewcptypes command 67
 viewer
 running for C API 33
 running for Java API 32
 using for troubleshooting 100
 viewfield command 74
 viewgroup command 74
 viewhistory command 93
 viewissue command 74
 viewproject command 75, 93
 viewquery command 76
 viewsandbox command 94
 viewstate command 77
 viewtype command 77
 viewuser command 77

W

work item
 retrieving from responses 27
 retrieving results for 28

Product Notices

Copyright © 2001–2006 MKS Software Inc.; in Canada copyright owned by MKS Inc. All rights reserved.

U.S. Government Restricted Rights. The software and documentation provided to you (the "Software and Documentation") are "commercial items," developed exclusively at private expense, consisting of "commercial computer software" and "commercial computer software documentation" as such terms are defined in the applicable acquisition regulations. If you are the U.S. Government or any agency or department thereof (collectively referred to as the "Government"), the Software and Documentation are licensed hereunder (i) only as a commercial item and (ii) with only those rights as are granted to all other end users pursuant to the terms and conditions of this Agreement to the extent that such rights are consistent with this paragraph. You agree to notify MKS if this Agreement fails to meet the Government's minimum needs or is inconsistent with U.S. federal procurement law. If you are any agency of the Department of Defense of the Government, the following notice is given: The Software and Documentation are provided with RESTRICTED RIGHTS. You shall not use, duplicate or disclose the Software or Documentation in any way not specifically permitted by this Agreement or mandated by U.S. law. Manufacturer is MKS Software Inc., 410 Albert Street, Waterloo, Ontario Canada N2L 3V3.

This product contains Apache Jakarta Commons HttpClient software developed by The Apache Software Foundation (<http://www.apache.org/>) licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This product includes Apache Jakarta Commons Logging 1.0.3 software developed by The Apache Software Foundation (<http://www.apache.org/>). Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This product contains Apache Tomcat v. 5.0.28 software developed by The Apache Software Foundation (<http://www.apache.org/>). Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The Java Software technologies are Copyright © 1994-2000 Sun Microsystems, Inc. ("Sun"). All rights reserved. This product contains Concurrent Package Release 1.3.4 developed by Sun. Neither the name of or trademarks of Sun may be used to endorse or promote products including or derived from the Java Software technology without specific prior written permission. This software is provided "AS IS," without a warranty of any kind. ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE

HEREBY EXCLUDED. SUN AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This product includes the Extreme! Computing XPP3-1.1.3.2 software developed by the Indiana University Extreme! Lab (<http://www.extreme.indiana.edu/>). Copyright (c) 2002 Extreme! Lab, Indiana University. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Indiana University Extreme! Lab (<http://www.extreme.indiana.edu/>)."; Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. 4. The names "Indiana University" and "Indiana University Extreme! Lab" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact <http://www.extreme.indiana.edu/>. 5. Products derived from this software may not use "Indiana University" name nor may "Indiana University" appear in their name, without prior written permission of the Indiana University. THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS, COPYRIGHT HOLDERS OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product contains Glazed Lists software developed by publicobject.com and O'Dell Engineering. Copyright (c) 2003-2005, publicobject.com, O'Dell Engineering Ltd. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. Copyright © 1991, 1999 Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

This product contains Bean Scripting Framework Version 2.2 developed by The Apache Software Foundation (<http://www.apache.org/>) licensed under the Apache License, Version 1.1. Copyright (c) [] The Apache Software Foundation. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>)."; 4. The names "BSF" "Apache" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org. 5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation. THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright © up to and including 2006, International Business Machines Corporation ("IBM") and others. All Rights Reserved. This product contains IBM Toolbox for Java software developed by IBM and is licensed under the IBM Public License Version 1.0 (the "License"); you may not use this file except in compliance with the License. See the License at <http://www-128.ibm.com/developerworks/library/os-ipl.html> for the specific language governing permissions and limitations under the License. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. IN NO EVENT SHALL THE CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product contains ICU v. 1.8.1. Copyright (c) 1995-2002 International Business Machines Corporation and others. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use,

copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder. All trademarks and registered trademarks mentioned herein are the property of their respective owners.

This product contains Java BEEP Core Version 0.9.08 developed by The Apache Software Foundation (<http://www.apache.org/>) licensed under the Apache License, Version 1.1. Copyright (c) 2000 The Apache Software Foundation. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>)."; 4. The names "Apache" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org. 5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation. THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product contains Java BEEP Core Version 0.9.08. Blocks Public License Copyright © 2000-2001, Invisible Worlds, Inc. All Rights Reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name, trademarks, or trade names of Invisible Worlds, Inc., nor the

names, trademarks, or trade names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL INVISIBLE WORLDS OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Cryptix General License Copyright © 1995, 1997, 1998, 2000. The Cryptix Foundation Limited. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistribution of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. THIS SOFTWARE IS PROVIDED BY THE CRYPTIX FOUNDATION LIMITED AND CONTRIBUTORS "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE CRYPTIX FOUNDATION LIMITED OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This package is a SSLv3/TLS implementation written by Eric Rescorla (ekr@rtfm.com) and licensed by Claymore Systems, Inc. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by Claymore Systems, Inc. Neither the name or Claymore Systems, Inc. nor the name of Eric Rescorla may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS

INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product contains Java Service Wrapper software developed by Tanuki Software. Copyright (c) 1999, 2005 Tanuki Software Inc. Permission is hereby granted, free of charge, to any person obtaining a copy of the Java Service Wrapper and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sub-license, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2001 Silver Egg Technology. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

This product contains JavaScript for Java which is subject to the Netscape Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.mozilla.org/NPL/>. Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License. The Original Code is Rhino code, released May 6, 1999. The Initial Developer of the Original Code is Netscape Communications Corporation. Portions created by Netscape are Copyright (C) 1997-1999 Netscape Communications Corporation. All Rights Reserved. Contributor(s): Norris Boyd

This product contains JBoss version 4.0.0. Copyright © 2004-2006 JBoss, Inc. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. Copyright © 1991, 1999 Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

This product contains Jdesktop Integration Components. Copyright © 2004 Sun Microsystems, Inc. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. Copyright © 1991, 1999 Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA. This library is distributed in the hope that it

will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

This product contains JFree Chart version 1.0.0. Copyright 2000-2006, by Object Refinery Limited and Contributors. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. Copyright © 1991, 1999 Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

This product contains Xerces 2.6.0 developed by The Apache Software Foundation (<http://www.apache.org/>) licensed under the Apache License, Version 1.1. Copyright (c) 2000-2002 The Apache Software Foundation. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by The Apache Software Foundation (<http://www.apache.org/>)."; 4. The names "Xerces", "Apache" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org. 5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation. THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product contains jTDS Drivers version 1.2 for MS SQL Server. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. Copyright © 1991, 1999 Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the

GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Copyright (c) 1998, 1999, JXML, Inc. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. All product materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by JXML, Inc. and its contributors: <http://www.jxml.com/mdsax/contributors.html> Neither name of JXML nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY JXML, INC. AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JXML OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product contains Log4j software licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This product contains NSPR Library software. The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.mozilla.org/MPL/>. Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License. The Original Code is the Netscape Portable Runtime (NSPR). The Initial Developer of the Original Code is Netscape Communications Corporation. Portions created by Netscape are Copyright (C) 1998-2000 Netscape Communications Corporation. All Rights Reserved.

This product contains OpenSSL software and is licensed under the terms of the OpenSSL License and the SSLeay License. Copyright (c) 1998-2005 The OpenSSL Project. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)" 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to

endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org. 5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project. 6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)" THIS SOFTWARE IS PROVIDED BY THE OPENSSL PROJECT "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OPENSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com). Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com). All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com). Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)" The word 'cryptographic' can be left out if the routines from the library being used are not cryptographic related :-). 4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: "This product includes software written by Tim Hudson (tjh@cryptsoft.com)" THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product contains Sapia Ubik software developed by Sapia Open Source Software. Copyright (c) 2002, 2003 Sapia Open Source Software. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following

disclaimer in the documentation and/or other materials provided with the distribution. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by Sapia Open Source Software Organization (<http://www.sapia-oss.org/>)."; Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. The names "Sapia", "Sapia Open Source Software" and "Sapia OSS" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact info@sapia-oss.org. Products derived from this software may not be called "Sapia", nor may "Sapia" appear in their name, without prior written permission of Sapia OSS. THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE SAPIA OSS ORGANIZATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product contains JDOM software. Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution. 3. The name "JDOM" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact request_AT_jdom_DOT_org. 4. Products derived from this software may not be called "JDOM", nor may "JDOM" appear in their name, without prior written permission from the JDOM Project Management request_AT_jdom_DOT_org. In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgement equivalent to the following: "This product includes software developed by the JDOM Project (<http://www.jdom.org/>)."; Alternatively, the acknowledgment may be graphical using the logos available at <http://www.jdom.org/images/logos>. THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Portions of this product are Copyright 2006, Microsoft Corporation. All rights reserved.

This product contains W3C Sirpac. This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions. Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications: The full text of this NOTICE in a location viewable to users of the redistributed or derivative work. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code. Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URLs to the location from which the code is derived.) THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS" AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS. COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION. The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

This product includes XML Parsing Library for C version 2.6.7. Copyright (C) 1998-2003 Daniel Veillard. All Rights Reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE DANIEL VEILLARD BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This product contains the Sun Java 1.5.0 Update Platform which may include the following software:

The following software may be included in this product: CS CodeViewer v1.0; Use of any of this software is governed by the terms of the license below: Copyright 1999 by CoolServlets.com. This software is distributed under the terms of the BSD License. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither name of CoolServlets.com nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS

PROVIDED BY COOLSERVLETS.COM AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE."

The following software may be included in this product: Crimson v1.1.1; Use of any of this software is governed by the terms of the license below: / The Apache Software License, Version 1.1 Copyright (c) 1999-2000 The Apache Software Foundation. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. 4. The names "Crimson" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org. 5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation. THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following software may be included in this product: Xalan J2; Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0> Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The following software may be included in this product: NSIS 1.0j; Use of any of this software is governed by the terms of the license below: Copyright (C) 1999-2000 Nullsoft, Inc. This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software. Permission is granted to anyone to use this software for any purpose, including

commercial applications, and to alter it and redistribute it freely, subject to the following restrictions: 1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required. 2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software. 3. This notice may not be removed or altered from any source distribution. Justin Frankel justin@nullsoft.com" Some Portions licensed from IBM are available at: <http://oss.software.ibm.com/icu4j/> Portions Copyright Eastman Kodak Company 1992 Lucida is a registered trademark or trademark of Bigelow & Holmes in the U.S. and other countries. Portions licensed from Taligent, Inc.

The following software may be included in this product:IAIK PKCS Wrapper; Use of any of this software is governed by the terms of the license below: Copyright (c) 2002 Graz University of Technology. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by IAIK of Graz University of Technology." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. 4. The names "Graz University of Technology" and "IAIK of Graz University of Technology" must not be used to endorse or promote products derived from this software without prior written permission. 5. Products derived from this software may not be called "IAIK PKCS Wrapper", nor may "IAIK" appear in their name, without prior written permission of Graz University of Technology. THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE LICENSOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following software may be included in this product: Document Object Model (DOM) v. Level 3; Use of any of this software is governed by the terms of the license below: W3Cýý SOFTWARE NOTICE AND LICENSE <http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>

This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions. Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications: 1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work. 2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code. 3. Notice

of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.) THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS. COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION. The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders. This formulation of W3C's notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". Otherwise, this version is the same as the previous version and is written so as to preserve the Free Software Foundation's assessment of GPL compatibility and OSI's certification under the Open Source Definition. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to site-policy@w3.org.

The following software may be included in this product: Xalan, Xerces; Use of any of this software is governed by the terms of the license below: / The Apache Software License, Version 1.1 Copyright (c) 1999-2003 The Apache Software Foundation. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. 4. The names "Xerces" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org. 5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation. THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following software may be included in this product: W3C XML Conformance Test Suites v. 20020606; Use of any of this software is governed by the terms of the license below: W3C's SOFTWARE NOTICE AND LICENSE Copyright © 1994-2002 World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved. <http://www.w3.org/Consortium/Legal/> This W3C work (including software, documents, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions: Permission to use, copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications, that you make: 1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work. 2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, a short notice of the following form (hypertext is preferred, text is permitted) should be used within the body of any redistributed or derivative code: "Copyright © [date-of-software] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved. <http://www.w3.org/Consortium/Legal/>" 3. Notice of any changes or modifications to the W3C files, including the date changes were made. (We recommend you provide URLs to the location from which the code is derived.) THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS. COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION. The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders. This formulation of W3C's notice and license became active on August 14 1998 so as to improve compatibility with GPL. This version ensures that W3C software licensing terms are no more restrictive than GPL and consequently W3C software may be distributed in GPL packages. See the older formulation for the policy prior to this date. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to site-policy@w3.org.

The following software may be included in this product: W3C XML Schema Test Collection v. 1.16.2; Use of any of this software is governed by the terms of the license below: W3C's DOCUMENT NOTICE AND LICENSE Copyright © 1994-2002 World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved. <http://www.w3.org/Consortium/Legal/> Public documents on the W3C site are provided by the copyright holders under the following license. The software or Document Type Definitions (DTDs) associated with W3C specifications are governed by the Software Notice. By using and/or copying this document, or the W3C document from which this statement is linked, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions: Permission to use, copy, and distribute the contents of this document, or the W3C document from which this statement is linked, in any medium for any purpose and without fee or royalty is hereby granted,

provided that you include the following on ALL copies of the document, or portions thereof, that you use: 1. A link or URL to the original W3C document. 2. The pre-existing copyright notice of the original author, or if it doesn't exist, a notice of the form: "Copyright © [date-of-document] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved. <http://www.w3.org/Consortium/Legal/>" (Hypertext is preferred, but a textual representation is permitted.) 3. If it exists, the STATUS of the W3C document. When space permits, inclusion of the full text of this NOTICE should be provided. We request that authorship attribution be provided in any software, documents, or other items or products that you create pursuant to the implementation of the contents of this document, or any portion thereof. No right to create modifications or derivatives of W3C documents is granted pursuant to this license. However, if additional requirements (documented in the Copyright FAQ) are satisfied, the right to create modifications or derivatives is sometimes granted by the W3C to individuals complying with those requirements. THIS DOCUMENT IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS. COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF. The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to this document or its contents without specific, written prior permission. Title to copyright in this document will at all times remain with copyright holders. This formulation of W3C's notice and license became active on April 05 1999 so as to account for the treatment of DTDs, schema's and bindings. See the older formulation for the policy prior to this date. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to site-policy@w3.org. webmaster (last updated by reagle on 1999/04/99.)

The following software may be included in this product: Common Unix Printing System API Libraries (CUPS API library);

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public

License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details. You should have received a copy of the GNU Library General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

The following software may be included in this product: Mesa 3-D graphics library v. 5; Use of any of this software is governed by the terms of the license below: core Mesa code include/GL/gl.h Brian Paul Mesa GLX driver include/GL/glx.h Brian Paul Mesa Ext registry include/GL/glxext.h SGI SGI Free B include/GL/glxext.h Mesa license: The Mesa distribution consists of several components. Different copyrights and licenses apply to different components. For example, GLUT is copyrighted by Mark Kilgard, some demo programs are copyrighted by SGI, some of the Mesa device drivers are copyrighted by their authors. See below for a list of Mesa's components and the copyright/license for each. The core Mesa library is licensed according to the terms of the XFree86

copyright (an MIT-style license). This allows integration with the XFree86/DRI project. Unless otherwise stated, the Mesa source code and documentation is licensed as follows: Copyright (C) 1999-2003 Brian Paul All Rights Reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL BRIAN PAUL BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. SGI FREE SOFTWARE LICENSE B (Version 1.1 [02/22/2000]) Except to the extent portions of this file are made subject to an alternative license as permitted in the SGI Free Software License B, Version 1.1 (the "License"), the contents of this file are subject only to the provisions of the License. You may not use this file except in compliance with the License. You may obtain a copy of the License at Silicon Graphics, Inc., attn: Legal Services, 1600 Amphitheatre Parkway, Mountain View, CA 94043-1351, or at: <http://oss.sgi.com/projects/FreeB> Note that, as provided in the License, the Software is distributed on an "AS IS" basis, with ALL EXPRESS AND IMPLIED WARRANTIES AND CONDITIONS DISCLAIMED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. Original Code. The Original Code is: [name of software, version number, and release date], developed by Silicon Graphics, Inc. The Original Code is Copyright (c) [dates of first publication, as appearing in the Notice in the Original Code] Silicon Graphics, Inc. Copyright in any portions created by third parties is as indicated elsewhere herein. All Rights Reserved. Additional Notice Provisions: [such additional provisions, if any, as appear in the Notice in the Original Code under the heading "Additional Notice Provisions"]

The following software may be included in this product: Byte Code Engineering Library (BCEL) v. 5; Use of any of this software is governed by the terms of the license below: Apache Software License / The Apache Software License, Version 1.1 Copyright (c) 2001 The Apache Software Foundation. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. 4. The names "Apache" and "Apache Software Foundation" and "Apache BCEL" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org. 5. Products derived from this software may not be called "Apache", "Apache BCEL", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation. THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following software may be included in this product: Regexp, Regular Expression Package v. 1.2; Use of any of this software is governed by the terms of the license below: The Apache Software License, Version 1.1 Copyright (c) 2001 The Apache Software Foundation. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. 4. The names "Apache" and "Apache Software Foundation" and "Apache Turbine" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org. 5. Products derived from this software may not be called "Apache", "Apache Turbine", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation. THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following software may be included in this product: CUP Parser Generator for Java v. 0.10k; Use of any of this software is governed by the terms of the license below: CUP Parser Generator Copyright Notice, License, and Disclaimer Copyright 1996-1999 by Scott Hudson, Frank Flannery, C. Scott Ananian Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the names of the authors or their employers not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. The authors and their employers disclaim all warranties with regard to this software, including all implied warranties of merchantability and fitness. In no event shall the authors or their employers be liable for any special, indirect or consequential damages or any damages whatsoever resulting

from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

The following software may be included in this product: JLex: A Lexical Analyzer Generator for Java v. 1.2.5; Use of any of this software is governed by the terms of the license below: JLEX COPYRIGHT NOTICE, LICENSE AND DISCLAIMER. Copyright 1996-2003 by Elliot Joel Berk and C. Scott Ananian Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the name of the authors or their employers not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. The authors and their employers disclaim all warranties with regard to this software, including all implied warranties of merchantability and fitness. In no event shall the authors or their employers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software. Java is a trademark of Sun Microsystems, Inc. References to the Java programming language in relation to JLex are not meant to imply that Sun endorses this product.

The following software may be included in this product: SAX v. 2.0.1; Use of any of this software is governed by the terms of the license below: Copyright Status SAX is free! In fact, it's not possible to own a license to SAX, since it's been placed in the public domain. No Warranty Because SAX is released to the public domain, there is no warranty for the design or for the software implementation, to the extent permitted by applicable law. Except when otherwise stated in writing the copyright holders and/or other parties provide SAX "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of SAX is with you. Should SAX prove defective, you assume the cost of all necessary servicing, repair or correction. In no event unless required by applicable law or agreed to in writing will any copyright holder, or any other party who may modify and/or redistribute SAX, be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use SAX (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or

a failure of the SAX to operate with any other programs), even if such holder or other party has been advised of the possibility of such damages. Copyright Disclaimers This page includes statements to that effect by David Megginson, who would have been able to claim copyright for the original work. SAX 1.0 Version 1.0 of the Simple API for XML (SAX), created collectively by the membership of the XML-DEV mailing list, is hereby released into the public domain. No one owns SAX; you may use it freely in both commercial and non-commercial applications, bundle it with your software distribution, include it on a CD-ROM, list the source code in a book, mirror the documentation at your own web site, or use it in any other way you see fit. David Megginson, sax@megginson.com 1998-05-11 SAX 2.0 I hereby abandon any property rights to SAX 2.0 (the Simple API for XML), and release all of the SAX 2.0 source code, compiled code, and documentation contained in this distribution into the Public Domain. SAX comes with NO WARRANTY or guarantee of fitness for any purpose. David Megginson, david@megginson.com 2000-05-05.

The following software may be included in this product: Cryptix; Use of any of this software is governed by the terms of the license below: Cryptix General License Copyright © 1995-2003 The Cryptix Foundation Limited. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. THIS SOFTWARE IS PROVIDED BY THE CRYPTIX FOUNDATION LIMITED AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE CRYPTIX FOUNDATION LIMITED OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.