

# Administrator's Guide

## **Citrix ICA Java Client**

Version 6.30

Citrix Systems, Inc.

Information in this document is subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Citrix Systems, Inc.

© 1994-2002 Citrix Systems, Inc. All rights reserved.

Citrix, Independent Computing Architecture (ICA), and MetaFrame are registered trademarks or trademarks of Citrix Systems, Inc. in the U.S.A. and other countries.

RSA Encryption © 1995-2000 RSA Security Inc, All Rights Reserved.

© 1999 Certicom Corp. All Rights Reserved.

© 1997-1998 Sun Microsystems, Inc. All Rights Reserved.

Java, JavaSoft, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Microsoft, MS, MS-DOS, Windows, Windows 2000, Windows ME, Internet Explorer, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

UNIX is a registered trademark of The Open Group in the U.S.A. and other countries.

Linux is a registered trademark of Linus Torvalds

Apple, Mac, Macintosh, MacOS, MacOS X, and MRJ are trademarks of Apple Computer, Inc., registered in the U.S. and other countries.

Netscape and the Netscape N and Ship's Wheel logos are registered trademarks of Netscape Communications Corporation in the U.S. and other countries. Netscape Navigator and Netscape Communicator are also trademarks of Netscape Communications Corporation and may be registered outside the U.S.

SOCKS5 is a registered trademark NEC Corporation.

Last updated: 28th March 2002 (PS)

# Contents

Before you Begin . . . . .	7
How to Use This Guide . . . . .	7
Citrix on the World Wide Web . . . . .	9
Providing Feedback About this Guide . . . . .	10

## Chapter 1

### Introduction to the Citrix ICA Java Client

Overview . . . . .	11
New Features in this Release . . . . .	11
Applet-mode Only Client Package . . . . .	11
ICA Settings Dialog Box . . . . .	12
Improved Performance . . . . .	12
TLS Encryption . . . . .	12
Citrix Secure Gateway . . . . .	12
Enhanced Internet Proxy Server Support . . . . .	12
Advanced Auto-reconnect . . . . .	13
NFuse Classic Integration . . . . .	13
User-to-User Shadowing . . . . .	13
Existing Features . . . . .	13
Introduction to the ICA Java Client . . . . .	14
ICA Java Client Requirements . . . . .	15

## Chapter 2

### Deploying the Citrix ICA Java Client

Tasks to Complete . . . . .	17
Deployment . . . . .	18
Unpacking the ICA Java Client . . . . .	18
Getting Started with the Sample HTML Files . . . . .	20
Using Signed Java Applets . . . . .	23

**Chapter 3****Configuring the Citrix ICA Java Client**

How to Configure the ICA Java Client . . . . .	26
Initial Configuration . . . . .	27
Starting the ICA Java Client in a Specific National Language . . . . .	27
Configuring Network Protocol and Server Location . . . . .	28
Setting the Client Name using the ICA Java Client . . . . .	30
Extended Parameter Passing . . . . .	31
User Interface Configuration . . . . .	32
Window Properties . . . . .	32
Status Bar and Settings Button . . . . .	32
Auto-reconnect and Session Termination . . . . .	33
Keyboard and Mouse . . . . .	34
Hotkeys . . . . .	37
Euro Currency Symbol Support . . . . .	39
Client Device Mapping . . . . .	39
Client Drive Mapping . . . . .	39
Client Printer Mapping . . . . .	40
COM Port Mapping . . . . .	43
Client Audio Mapping . . . . .	44
Improving Performance . . . . .	46
Data Compression . . . . .	46
Bitmap Cache . . . . .	46
Queuing Mouse Movements . . . . .	47
Using SpeedScreen Latency Reduction . . . . .	47
Improving Performance Over a Low-bandwidth Connection . . . . .	48
Integrating the ICA Java Client with Security Solutions . . . . .	50
Connecting Through a Proxy Server . . . . .	50
Integrating the ICA Java Client with Citrix Secure Gateway or SSL Relay . . . . .	53
Connecting to a Server Across a Firewall . . . . .	57
Advanced Configuration . . . . .	59
Precedence of Configuration Methods . . . . .	59
Copying Client Drive Mapping Configuration . . . . .	60
Changing the List of Printer Drivers that Appears in the ICA Java Client Settings dialog box . . . . .	61

---

ICA Java Client Parameters .....	63
General Configuration .....	63
User Interface Parameters .....	65
Client Device Mapping Parameters .....	67
Performance Tuning Parameters .....	67
Security Integration Parameters .....	68
Troubleshooting .....	68
Server names with non-ASCII characters .....	69
Client Drive Mapping .....	69
Client Printer Mapping .....	70
Client Audio Mapping .....	70
Clipboard support on X11 .....	71
Mouse Pointer Support .....	71
Internet Explorer .....	71
Netscape Navigator .....	72
MacOS 9.x .....	72
Miscellaneous .....	73
<b>Index .....</b>	<b>75</b>



---

# Before you Begin

This manual is for system administrators responsible for installing, configuring, and maintaining the Citrix ICA Client for Java (also called the ICA Java Client). This manual assumes knowledge of:

- The MetaFrame server to which your ICA Clients connect
- The operating system on the client computer
- Java-enabled Web browsers, including HTML and JavaScript
- Installation, operation, and maintenance of network and printing systems

## How to Use This Guide

To get the most out of the *Citrix ICA Java Client Administrator's Guide*, review the table of contents to familiarize yourself with the topics discussed.

This guide contains the following sections:

<b>Chapter</b>	<b>Contents</b>
Chapter 1, "Introduction to the Citrix ICA Java Client"	Gives a list of features and system requirements.
Chapter 2, "Deploying the Citrix ICA Java Client"	Describes how to deploy the ICA Java Client.
Chapter 3, "Configuring the Citrix ICA Java Client"	Describes how to start and configure connections to MetaFrame servers.

# Conventions

The following conventional terms, text formats, and symbols are used throughout the printed documentation:

Convention	Meaning
<b>Bold</b>	Indicates boxes and buttons, column headings, command-line commands and options, dialog box titles, lists, menu names, tabs, and menu commands.
<i>Italic</i>	Indicates a placeholder for information or parameters that you must provide. For example, if the procedure asks you to type <i>filename</i> , you must type the actual name of a file. Italics also indicate new terms and the titles of other books.
ALL UPPERCASE	Represents keyboard keys (for example, CTRL, ENTER, F2).
[brackets]	Encloses optional items in syntax statements. For example, [password] indicates that you can choose to type a password with the command. Type only the information within the brackets, not the brackets themselves.
...(ellipsis)	Indicates a command element that can be repeated.
Monospace	Represents examples of screen text or entries that you might type at the command line or initialization files.
▶	Indicates a procedure.
▪	Indicates a list of related information, not procedural steps.

The Citrix ICA Java Client allows users to connect to MetaFrame servers. When describing a feature unique to a particular MetaFrame server, this manual specifies the appropriate server and version number.

---

**Note** This manual uses the word Windows to describe Microsoft Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, and Windows ME operating systems unless otherwise stated.

---



## Finding More Information

This manual contains conceptual information and installation and configuration steps for the ICA Java Client. For additional information, consult the following:

- The online help for the ICA Client you deploy
- The *Citrix ICA Client Administrator's Guides* for the other ICA Clients you deploy
- The documentation included in your MetaFrame server package for instructions about installing, configuring, and maintaining your MetaFrame servers

This book and other Citrix documentation is available in Adobe PDF format in the following locations:

- The documentation directory of your MetaFrame XP Components CD-ROM disc
- <http://www.citrix.com/download/> Click the ICA Client platform for which you want information
- <http://www.citrix.com/support/> Click Product Documentation and choose ICA Client

Using the Adobe Acrobat Reader, you can view and search the documentation electronically or print it for easy reference. You can download the Adobe Acrobat Reader for free from the Adobe Web site at <http://www.adobe.com>

---

**Important** Always consult the Readme files for your MetaFrame server and the Citrix ICA Client for any last-minute updates, installation instructions, and corrections to the documentation.

---

## Citrix on the World Wide Web

The Citrix Web site, at <http://www.citrix.com/>, offers a variety of information and services for Citrix customers and users. From the Citrix home page, you can access Citrix online Technical Support Services and other information designed to assist MetaFrame administrators, including the following:

- Downloadable Citrix ICA Clients at <http://www.citrix.com/download/>
- Citrix Product Documentation Library, containing the latest documentation for all Citrix products (at <http://www.citrix.com/support/> select Product Documentation). You can download updated editions of the documentation that ships with Citrix products and supplemental documentation that may only be available from the Citrix Web site.

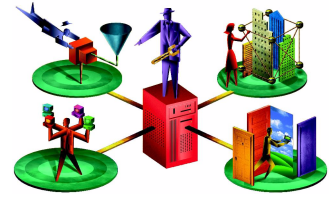
- Program information about Citrix Preferred Support Services options
- An FTP server containing the latest service packs, hotfixes, utilities, and product literature for download
- An online Solution Knowledge Base containing an extensive collection of application notes, technical articles, troubleshooting tips, and white papers
- Interactive online Solution Forums for discussion of technical issues with other users
- Frequently Asked Questions pages with answers to common technical and troubleshooting questions
- Information about programs and courseware for Citrix training and certifications
- Contact information for Citrix headquarters, including worldwide, European, Asia Pacific, and Japan headquarters
- The Citrix Developer Network (CDN) at <http://www.citrix.com/cdn/> This open enrollment membership program provides access to developer tool kits, technical information, and test programs, for software and hardware vendors, system integrators, ICA licensees, and corporate IT developers who incorporate MetaFrame server-based computing solutions into their products.

## Providing Feedback About this Guide

We strive to provide accurate, clear, complete, and usable documentation for Citrix products. If you have any comments, corrections, or suggestions for improving our documentation, we want to hear from you.

You can send email to the documentation authors at [documentation@citrix.com](mailto:documentation@citrix.com). Please include the product name and version number, and the title of the document in your message.

# Introduction to the Citrix ICA Java Client



## Overview

The ICA Java Client is a Java applet that provides access to applications running on a Citrix MetaFrame server farm from anywhere with a standard Web browser. The applet is a download-and-run, zero-install client, optimized for using in Web environments where it is not possible or desirable to install software on the client device.

## New Features in this Release

The following features are new in this release:

- Applet-mode only client package, available in .zip and .tar.gz formats
- ICA Settings dialog box and status bar
- Improved performance and reduced bandwidth usage
- Support for TLS Encryption
- Support for Citrix Secure Gateway 1.1
- Enhanced Internet proxy support
- Auto client reconnect

## Applet-mode Only Client Package

The ICA Java Client has now been optimized and streamlined for use in environments where access to Windows and UNIX applications through a Web browser is required. It is no longer available as an installable alternative to native clients because Citrix provides an excellent range of native clients covering virtually all platforms.

Unlike previous releases, the ICA Java Client is now supplied as an applet-mode only client package. The package is available to download from <http://www.citrix.com> in two formats: .zip and .tar.gz.

## ICA Settings Dialog Box

The ICA settings can now be configured by the end-user using a pop-up dialog in their Web browser. If allowed by the administrator, they can store persistent settings on their local desktop. In particular, printer configuration is now far easier, with a list of the most common printer types provided in the dialog box.

## Improved Performance

The ICA Java Client is extensively optimized for deployment from a Web browser, as a zero-install, rapid-download client that can run anywhere. Tight integration with NFuse Classic 1.7 includes the ability for the administrator to specify which components should be deployed. Speed has increased by as much as 100% over previous versions.

## TLS Encryption

As an alternative to SSL 3.0, the ICA Java Client also supports TLS 1.0. TLS (Transport Layer Security) is the standardized form of SSL (Secure Sockets Layer). Both are cryptographic security protocols designed to ensure the integrity and privacy of data transfers across public networks. SSL and TLS are functionally equivalent.

## Citrix Secure Gateway

The ICA Java Client provides full support for Citrix Secure Gateway Version 1.1. Citrix Secure Gateway acts as a secure Internet gateway between SSL/TLS-enabled ICA Clients and MetaFrame XP servers. The Internet portion of ICA traffic between client devices and the Citrix Secure Gateway server is SSL/TLS-encrypted. SSL/TLS is the Internet standard 128-bit encryption technology used for client-server authentication, and to ensure the integrity and privacy of data transfers across public networks.

## Enhanced Internet Proxy Server Support

As an alternative to using a SOCKS proxy, the ICA Java Client also supports using a Secure proxy (also known as Security proxy, HTTPS proxy, and SSL-tunnelling). Proxy authentication is also supported. When used with Citrix Secure Gateway, applications can be delivered securely to anywhere in the world, via the Internet.

The ICA Java Client can automatically detect proxy server settings from the Web browser.

## Auto Client Reconnect

ICA sessions can be dropped for a variety of reasons, including unreliable networks, highly variable network latency, or range limitations of wireless devices. The *auto client reconnect* feature is triggered when the ICA Client detects a disconnected session. When this feature is enabled on a MetaFrame XP server, users can transparently reconnect to their session without having to re-enter their logon credentials. Automatic client reconnection does not occur if users exit applications without logging off.

## NFuse Classic Integration

A new interface in NFuse Classic allows the administrator to automatically configure many properties of the ICA Java Client. It can optionally allow the end user to select the functionality they require from the client to minimize download times. See the *NFuse Classic Administrator's Guide* for more information.

## User-to-User Shadowing

Shadowing is the process of monitoring a user's session remotely and, optionally, participating in the session using your own keyboard and mouse. Previously reserved for MetaFrame administrators, this feature no longer requires administrative rights. You can now make shadowing available to users, adding powerful collaborative capabilities to a variety of applications, including

- Help desks, where trained personnel troubleshoot applications
- Training organizations, where students observe instructors, and instructors monitor student performance
- Presentation sessions, where multiple remote users attend a presentation hosted by a single user

Shadowing does not require client configuration. This feature is governed by user policies set on the MetaFrame XP server. For information about enabling and configuring this feature, see the *MetaFrame XP Administrator's Guide* included in your MetaFrame XP, Feature Release 2 media pack.

## Existing Features

When connected to a MetaFrame server, the ICA Java Client provides additional features that make remote computing just like running applications on a local desktop. The ICA Java Client has the following features:

- SSL encryption - SSL provides server authentication, encryption of the data stream, and message integrity checks.

- Extended Parameter Passing - This allows you to pass parameters to a published application.
- Video support
  - Resolutions up to 65536 x 65536
  - 256 color to 24-bit (16.7 million colors)
- Client clipboard mapping
- Client device mapping
  - Client printer mapping
  - Client drive mapping
  - Client audio mapping
  - COM port mapping
- ICA encryption
- Data compression
- Data caching
- SpeedScreen Latency Reduction
- Business Recovery
- Hotkeys
- Shadowing

## Introduction to the ICA Java Client

The ICA Java Client, also known as the *Java Web Client*, is a signed Java applet that allows users to connect to MetaFrame servers.

The ICA Java Client has the following advantages:

- **Zero client install** - you do not need to install any software on any client machine. Users only require a Java-compatible Web browser. Setup is transparent and automatic.
- **Minimal download size** - at the most basic level of functionality, the applet is around 280k in size, providing a faster download than any other ICA Client.
- **Platform-independent** - The ICA Java Client can run on any client hardware running a Java-compatible Web browser.

The ICA Java Client applet resides on a Web server and is deployed using an HTML page with the ICA Java Client applet tag. Users run the ICA Java Client by visiting the HTML page using a browser that has Java support. When users request the page, the Java applet is automatically downloaded to the client machine. The applet then runs and connects to the server or published application specified in the applet tag.

Applet mode sessions offer considerable control to the network administrator and are preferable when administrators want to relieve users of the mechanics of configuring and launching sessions. This high level of remote configuration is particularly advantageous when using Citrix NFuse Classic.

Unlike the ActiveX, Netscape plug-in, or Win32 Web Clients, which are downloaded once and then saved for future use by client systems, the Java applet is not stored permanently by the client system. However, most browsers cache Java applets so that they do not need to be downloaded each time they are used.

## ICA Java Client Requirements

To run the ICA Java Client, the client system must have the following:

- A Web browser with Java Runtime Environment 1.1, or Java 2, Standard Edition Version 1.3 or greater support
- TCP/IP (Dial-up or LAN)
- Mouse
- Keyboard
- Access to a Web server that will store the ICA Java Client files

### Java Environments

A large number of Java-enabled environments are available, and their functionality varies from platform to platform. To validate proper functionality of the ICA Java Client, Citrix selects a representative group of platforms for testing. Java 2, Standard Edition Version 1.2 is not recommended for use with the ICA Java Client. Upgrade to a Web browser with Java 2, Standard Edition Version 1.3 or upgrade the JVM in your Web browser to Java 2, Standard Edition Version 1.3. See <http://java.sun.com/> for more information about JVM upgrades.





# Deploying the Citrix ICA Java Client



This chapter explains how to deploy the ICA Java Client applet. You can also deploy the client with NFuse Classic, which automates the procedures described in this chapter. NFuse Classic is described in the *NFuse Classic Administrator's Guide*. We recommend that you investigate deploying the ICA Java Client with NFuse Classic as this automates many of the tasks described in this chapter.

## Tasks to Complete

In this chapter you will complete the following tasks:

- Deploy the ICA Java Client on a Web server and customize the ICA Java Client deployment
- Learn how to use Web browsers with signed Java applets

# Deployment

## Before you Begin

This chapter describes a basic deployment of the ICA Java Client applet, followed by details of how to customize the Java applet and how to pass parameters to the client. When deploying the ICA Java Client, you can control almost any aspect of the client configuration, as well as controlling what functionality will be available to the user, and how large the download will be.

To deploy the ICA Java Client, you need:

- A copy of the client package. The ICA Java Client package is available in two formats, .zip, primarily for Windows systems and .tar.gz, primarily for UNIX systems. Both contain identical contents. The package can be downloaded from <http://www.citrix.com/download> or copied in uncompressed form from the Components CD. Citrix recommends that you obtain the latest version of the client from the website.

---

**Note** For evaluation purposes, a minimal, quick-download version of the client package is available. It contains only the necessary archives to create an ICA connection. It is available in .zip and .tar.gz formats.

---

- A third-party application to uncompress and unpack the .zip or .tar.gz package
- Administrator access to a Web server

## Unpacking the ICA Java Client

This section describes the contents of the ICA Java Client package

### ► To unpack the ICA Java Client

1. Copy the ICA Java Client package to a suitable location on your Web server. For Microsoft IIS servers, copy the package to a folder in the Web Root directory. For UNIX systems, consult your Web server documentation.
2. Extract the program files from the .zip or .tar.gz package using a suitable decompression utility.

When you extract the client package, a number of files are created on your Web server. Some of these files are the Java archives that make up the ICA Java Client applet. There are three types of archive package, as follows:

Archive type	Description
*M.cab files	Authenticode signed cabinet files. Compatible with: Internet Explorer 4.x or later in Windows systems
*N.jar files	Netscape Object signed Java archives. Compatible with: Netscape Navigator/Communicator 4.x or later (all platforms apart from MacOS 9.x) Netscape 6 (all platforms apart from MacOS 9.x or MacOS X) Internet Explorer on Windows platforms with Java Plugin* 1.3.1 or later
*J.jar files	Sun Javasoft signed Java archives. Compatible with: Internet Explorer on MacOS 9.x Internet Explorer on MacOS X Netscape Navigator/Communicator 4.76 or later on MacOS 9.x with MRJ plugin Netscape 6 (all platforms apart from MacOS X) Internet Explorer on Windows platforms with Java Plugin* 1.3.1 or later

\* Download the Java Plug-in from <http://java.sun.com>

For each type of signed archive package there are a number of different components. For Netscape Object signed Java archives, the components are named as follows.

Archive File	Description
JICAEngN.jar	Complete archive. This archive contains the contents of all of the other archives apart from cryptojN.jar and sslN.jar which must be included if required.
cryptojN.jar	Encryption component. Required for all encryption.
sslN.jar	SSL component. Adds SSL and TLS encryption support.
JICA-configN.jar	User configuration component. Adds the Status bar, buttons, and ICA Settings dialog box.
JICA-coreN.jar	Core archive. This archive provides only a basic connection. You add functionality by using it in conjunction with the other component archives described below.
JICA-audioN.jar	Audio component. Adds client audio mapping.
JICA-cdmN.jar	CDM component. Adds client drive mapping.
JICA-clipboardN.jar	Clipboard component. Adds client clipboard support.

JICA-commN.jar	Communications component. Adds serial port mapping support. Also requires the javax.comm standard extension package.
JICA-printerN.jar	Printer component. Adds client printer mapping.
JICA-sicaN.jar	ICA Encryption component. Adds ICA Encryption Support.
JICA-tw1N.jar	Thinwire 1 component. Required for connections to MetaFrame 1.8 and MetaFrame 1.8 SP1 servers.
JICA-zlcN.jar	ZLC component. Adds SpeedScreen Latency Reduction support.

---

If you are not using Netscape Object signed Java archives, choose the appropriate archive for the JVM you are using. For example, to deploy the core archive for Internet Explorer, choose the Authenticode signed archive, JICA-coreM.cab.

## Getting Started with the Sample HTML Files

For users to access the client and start an ICA session, you must create an HTML page that specifies the correct archives for the user's browser, or customize one of the sample HTML pages supplied in the package.

There are four HTML files supplied in the client package:

- **index.html.** This page contains links to and descriptions of the three launching pages, desktop.html, application.html, and autoproxy.html.
- **desktop.html.** This page launches a desktop session to a MetaFrame server. To make a connection with this page, you need to specify an address for the server.
- **application.html.** This page launches a connection with 128-bit encryption enabled, to a published application. You need to specify the name of the published application and the name of a MetaFrame server to use for server location.
- **autoproxy.html.** This page launches a connection to a published application through a proxy server, using auto client proxy detection. You need to specify the name of the published application and the name of a MetaFrame server to use for server location.

## Editing desktop.html

Desktop.html, contains the following applet tag:

```
<applet name="javaclient"
  codebase=".."
  code=com.citrix.JICA
  archive="JICA-coreN.jar,JICA-configN.jar"
  width=640
  height=480>
  <param name="cabinets" value="JICA-coreM.cab,JICA-configM.cab">
  <param name="Address" value="avalon">
  <param name="End" value="end.html">
</applet>
```

The applet tag is used to configure the ICA Java Client. Some parameters are specified inside the <applet> tag as follows:

- **Applet name.** This is an optional, unique name for the applet. Use this name to refer to the applet when writing scripts. In desktop.html, the applet name 'javaclient' is used to activate a script that displays a warning message if the user tries to close the Web browser window when an ICA session is running.
- **Codebase.** The relative path from the HTML page to the ICA Java Client archives. Change this path if it is not correct for your deployment.
- **Code.** The name of the class file that is executed. For the ICA Java Client, this is always com.citrix.JICA.
- **Archive.** Specify Netscape Object signed or JavaSoft signed archives here. Specify Authenticode (Internet Explorer) signed archives in a <param> tag. In desktop.html, a Netscape Object signed core archive is specified here and Authenticode signed archives are specified in the <param> tag so that both Netscape and Internet Explorer are supported. Multiple archives specified in the same place are separated by a comma.
- **Width.** The width of the ICA session window, in pixels.
- **Height.** The height of the ICA session window, in pixels.

All other parameters are specified using <param> tags, located between the <applet> and </applet> tags. The <param> tags are always used in the form,

```
<param name="parametername" value="valuename">
```

where *parametername* is the name of the parameter you are specifying and *valuename* is the value you are allocating to it.

---

**Note** You cannot combine Javasoft and Netscape Object signed archives in the same <applet> tag. To specify more than one type of .jar archive in the same HTML page, you must use a Web browser identification script.

---

► **To customize and use desktop.html**

1. Open desktop.html in a plain text HTML editor or text editor and find the <applet> tag section:

```
<applet name="javaclient"
  codebase=".."
  code=com.citrix.JICA
  archive="JICA-coreN.jar,JICA-configN.jar"
  width=640
  height=480>
  <param name="cabinets" value="JICA-coreM.cab,JICA-configM.cab">
  <param name="Address" value="avalon">
  <param name="End" value="end.html">
</applet>
```

This is the section that launches the ICA Java Client.

2. Change the Address value of avalon to the address of a MetaFrame server on your local network. If you are connecting to a published application, change the address to the name of the published application and add the initialprogram parameter. For example, if the published application is called notepad, use the following parameters.

```
<param name="address" value="notepad">
<param name="initialprogram" value="#notepad">
```

3. Change the relative path specified for the codebase if it is not correct for your deployment.
4. Publish the sample HTML pages using your Web server. See your Web server documentation for more information on how to do this.
5. Open a Web browser and open the URL for the sample HTML pages. The index.html page opens.
6. Click the **Minimal Desktop** link. The ICA Java Client applet appears:

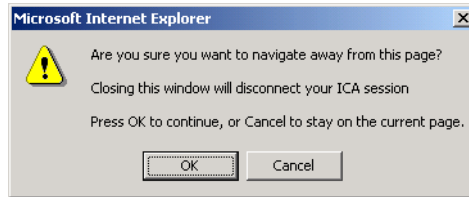


7. Click **Connect** or **Click to connect** to connect to your MetaFrame server. Click **Settings** to display the ICA Java Client Settings dialog box.

You can edit `application.html` and `autoproxy.html` in the same way. The additional parameters used in these examples are described in chapter 3.

► **To change the warning message displayed when a user tries to close an active ICA session**

If you try to close the Web browser window when using an ICA session, created with one of the example Web pages, a warning message appears:



The message “Closing this window will disconnect your ICA session” is defined in this section of the HTML page:

```
function onBeforeUnload() {  
    var connected = document.javaclient.isConnected();  
    if (connected) {  
        alerted = true;  
        return "Closing this window will disconnect your ICA  
session";  
    }  
}
```

- To change the message displayed, edit the text in the HTML page.

## Using Signed Java Applets

Due to network security restrictions imposed by Java, many Web browsers do not permit users to connect to other computers on the network when using Java applets.

When a Java applet attempts to make a connection to the server specified in the HTML page, the Java security manager detects the attempt to connect to another computer and cancels the operation. The result of this security restriction is that, under normal conditions, a client system can connect to a server only if the server is also the same machine as the Web server that contains the applet class files.

To overcome this restriction, the ICA Java Client uses signed archives. The signature confirms that the files being downloaded came from Citrix and have not been altered since the signature was applied. You must ensure that your Web browser is configured to accept signed Java applets.

When attempting a connection to the MetaFrame server, the user is prompted with the Citrix signed certificate. When the user accepts the signature, the connection is permitted.





# Configuring the Citrix ICA Java Client



This chapter describes how to configure and use the ICA Java Client:

- How to configure the ICA Java Client
- Initial configuration
  - Starting the ICA Java Client in a specific national language
  - Configuring network protocol and server location
  - Setting the client name with the ICA Java Client
  - Extended parameter passing
- User interface configuration
  - Window properties
  - Status bar and settings button
  - Auto-reconnect and session termination
  - Keyboard and mouse
  - Hotkeys
  - Euro currency symbol support
- Client device mapping
  - Client drive mapping
  - Client printer mapping
  - Client COM port mapping
  - Client audio mapping
- Improving performance

- Data compression
- Data caching
- Queuing mouse movements
- SpeedScreen latency reduction
- Improving performance over a low-bandwidth connection
- Integrating the ICA Java Client with security solutions
  - Proxy servers
  - CSG and SSL Relay
  - Firewalls
  - ICA encryption
- Advanced configuration
- ICA Java Client parameters
- Troubleshooting
  - Common errors

## How to Configure the ICA Java Client

The ICA Java Client can be configured:

- using an HTML page, as described in Chapter 2
  - using the **ICA Java Client Settings** dialog box
  - using NFuse Classic
- ▶ **To configure the ICA Java Client using the ICA Java Settings dialog box**
- To configure the client using the **ICA Java Settings** dialog box, click the **Settings** button on the status bar when the client is running in a Web browser window. The **ICA Java Client Settings** dialog box appears. Click the **Help** buttons to display context-sensitive help. Each of the features that can be configured from the dialog is also described in detail in this chapter.
- ▶ **To configure the ICA Java Client using NFuse Classic**
- NFuse Classic automatically generates the necessary Web pages to launch the ICA Java Client. See the *NFuse Classic Administrator's Guide* for details of how to use NFuse Classic.

# Initial Configuration

## Starting the ICA Java Client in a Specific National Language

The ICA Java Client allows users to specify what locale to use to display the user interface. By default, if you do not specify a locale, your session uses the locale of your client machine to display the user interface. If you specify a language code that is not recognized or not supported, English is used.

The following locales are supported by the ICA Java Client:

- English
- German
- French
- Spanish
- Japanese

### ► To set the ICA Java client locale

- Specify the following parameter in the HTML page:

```
<param name="language" value="yourlocale">
```

Where *yourlocale* is the two letter abbreviation for the locale you want to use. The standard two letter abbreviations are as follows:

**en** - English

**fr** - French

**de** - German

**es** - Spanish

**ja** - Japanese

For example, to use Japanese as your locale on a non-Japanese machine when connecting to the MetaFrame server named CitrixServer, create an applet tag like the following:

```
<applet code=com.citrix.JICA
  archive=JICAEngN.jar
  width=1024 height=768>
<param name="cabinets" value="JICAEngM.cab">
<param name="address" value="CitrixServer">
<param name="language" value="ja">
</applet>
```

## Configuring Network Protocol and Server Location

This section explains how to:

- Set the network protocol the ICA Java Client uses
- Control the way in which the ICA Java Client locates MetaFrame servers
- Configure a list of MetaFrame servers for Business Recovery

The network protocol setting allows you to control the way the ICA Client searches for MetaFrame servers and how it communicates with them.

The protocols are:

- **TCP/IP + HTTP** - The client uses the HTTP protocol to search for Citrix servers. The client communicates with the MetaFrame server using ICA protocol over TCP/IP. This is the default protocol.
- **SSL/TLS + HTTPS** - The client uses the HTTPS protocol to search for a list of MetaFrame servers. The client communicates with the MetaFrame server using the SSL or TLS protocols. These protocols are described in detail in “Integrating the ICA Java Client with Citrix Secure Gateway or SSL Relay” on page 53.
- **TCP/IP** - The client uses the UDP protocol to search for MetaFrame servers. The client communicates with the MetaFrame server using ICA protocol over TCP/IP. Select this option if all of the MetaFrame servers and clients are located on the same network and a secure connection is not required.

Server location (also called server browsing) provides a method for a network-connected ICA Client to connect to a published application. The way in which server location works depends on which network protocol has been configured:

- **TCP/IP+HTTP and SSL/TLS+HTTPS** - The default server address is *ica* or *ica.<current DNS domain>* depending on the behavior of your JVM. You must set specific server addresses for MetaFrame servers. The ICA client uses the HTTP or HTTPS protocol respectively, to contact the servers.
- **TCP/IP** - The default setting for server location is auto-locate. The ICA client attempts to contact all of the servers on the network by broadcasting on the UDP protocol. Alternatively, you can set specific addresses for MetaFrame servers.

Business recovery provides consistent connections to published applications in the event of a master ICA browser server disruption. You can define up to three groups of MetaFrame servers to which you want to connect: a primary and two backups. Each group can contain from one to five servers.

By default, the ICA Java Client uses the TCP/IP+HTTP protocol. You can change the protocol and specify Business Recovery server addresses in an HTML page as described here.

▶ **To change the network protocol to TCP/IP (UDP)**

- Add the following parameters to the HTML page:

```
<param name="BrowserProtocol" value="UDP">
```

▶ **To change the network protocol to SSL/TLS+HTTPS**

- Add the following parameters to the HTML page:

```
<param name="BrowserProtocol" value="HTTPOnTCP">  
<param name="SSLEnable" value="on">
```

▶ **To specify a business recovery server group using the TCP/IP+HTTP or SSL/TLS+HTTPS protocols**

Use the following parameters to specify server groups:

- **HTTPBrowserAddress to HTTPBrowserAddress5:** Specify the primary group of servers
- **HTTPBrowserAddress6 to HTTPBrowserAddress10:** Specify the first backup group of servers
- **HTTPBrowserAddress11 to HTTPBrowserAddress15:** Specify the second backup group of servers

Fill in any unused addresses with five dashes (-----). These dashes are required to fill in any gaps in the list but are not required at the end of the list.

▶ **To specify a business recovery server group using the TCP/IP protocol**

Use the following parameters to specify server groups:

- **TCPBrowserAddress to TCPBrowserAddress5:** Specify the primary group of servers
- **TCPBrowserAddress6 to TCPBrowserAddress10:** Specify the first backup group of servers
- **TCPBrowserAddress11 to TCPBrowserAddress15:** Specify the second backup group of servers

Fill in any unused addresses with five dashes (-----). These dashes are required to fill in any gaps in the list but are not required at the end of the list.

## Example

In the following example, the primary group of servers contains “Arthur”, “Morgana” and “Merlin”. The first backup group contains the servers “Excalibur” and “Stone”. There is no secondary backup group:

```
<html>
<body>
<applet code=com.citrix.JICA
archive=JICAEngN.jar
width=640 height=480>
<param name="Cabinets" value="JICAEngM.cab">
<param name="BrowserProtocol" value="HTTPonTCP">
<param name="HTTPBrowserAddress" value="Arthur">
<param name="HTTPBrowserAddress2" value="Morgana">
<param name="HTTPBrowserAddress3" value="Merlin">
<param name="HTTPBrowserAddress4" value="----">
<param name="HTTPBrowserAddress5" value="----">
<param name="HTTPBrowserAddress6" value="Excalibur">
<param name="HTTPBrowserAddress7" value="Stone">
</applet>
</body>
</html>
```

## Setting the Client Name using the ICA Java Client

The client name is used to identify the ICA Client to the MetaFrame server, and is also used to name auto-created printers. See “Client Printer Mapping” on page 40 for more information on auto-created printers. You may need to change the client name if you are trying to access resources shared with another ICA Client with the same client name.

By default, the Java Client uses the value for the client machine’s host name (if it exists and isn’t set to “localhost”) as the client name reported to the MetaFrame server. If the ICA Java Client cannot use the client machine’s host name, it uses “AnonJava” as the client name. The client name sent to the MetaFrame server is always truncated to 20 characters.

### ► To change the client name

- To change the client name, you need to add two parameters in the HTML Page:

```
<param name="client.wfclient.UseHostname" value="off">
<param name="client.wfclient.Clientname" value="yourclient">
```

where *yourclient* is the client you are using.

## Extended Parameter Passing

When connecting to a published application, you specify the program name using the `initialprogram` parameter. For example, to connect to an application called `notepad` specify the following parameters:

```
<param name="address" value="notepad" >
<param name="initialprogram" value="#notepad">
```

### ► To pass a parameter to an application

- Specify the following parameters in the HTML page:

```
<param name="address" value="application" >
<param name="initialprogram" value="#application" >
<param name="param" value="parameter" >
```

Where *application* is the name of the published application and *parameter* is the parameter. You can specify a maximum value length of 256 characters for these parameters. For example, to open a file called `M:\new.txt` in `notepad` specify the following parameters:

```
<param name="address" value="notepad" >
<param name="initialprogram" value="notepad" >
<param name="param" value="M:\new.txt" >
```

---

**Note** For extended parameter passing to work, the MetaFrame Administrator must configure the published application to receive parameters by appending `%*` to the published application's command line, for example:

```
notepad %*
```

Full details of how to publish applications and set up commands can be found in the MetaFrame Administrator's Guide.

---

Any filename parameters are interpreted by the remote application relative to the server's file system. If you want to pass a client-side file to a remote published application then you must use client drive mapping.

## User Interface Configuration

### Window Properties

You set the size of the window that your ICA session runs in using the applet tag. See “Deployment” on page 18 for more information. The number of colors used in the session window is defined with a parameter.

▶ **To set the number of colors used in the ICA session window**

- Specify the following parameter in the HTML page

```
<param name="DesiredColor" value="2|4|8" >
```

Where 2 specifies 256 colors, 4 specifies thousands of colors, and 8 specifies millions of colors. The ICA Java Client cannot be configured to use only 16 colors but can display applications published in 16 color mode; these will be run in 256 colors.

### Status Bar and Settings Button

The status bar displays status information about the client. When you click the **Settings** button, the **ICA Java Client Settings** dialog box appears.

You can display or hide the status bar and **Settings** button using parameters in the HTML page. Both are displayed by default.

▶ **To hide the status bar**

- Specify the following parameter in the HTML page:

```
<param name="ShowStatusBar" value="no" >
```

▶ **To hide the Settings button**

- Specify the following parameter in the HTML page:

```
<param name="ShowSettingsButton" value="no" >
```

---

**Note** To display the **Settings** button and status bar when using the core archives, you must include the JICA-config\*. \* archive. This functionality is included in the complete archives. If you have not included the necessary archives for client drive mapping or printer mapping, the corresponding tabs will not appear in the **ICA Java Client Settings** dialog box.

---



## Auto-reconnect and Session Termination

You can control how the client behaves when starting or ending a session.

When starting a session, you can configure the client to connect when you click the splash screen, or to connect automatically, as the HTML page displays. You can also change what happens when a session ends - either as a result of you terminating it, or because there was a problem.

### ► To change the ICA Java Client startup

- Specify the following parameter in the HTML page:

```
<param name="start" value="manual|auto">
```

If you set this parameter to **manual** (default), you must click to connect to a MetaFrame server. If you set it to **auto**, the message “Connecting to server” appears and you are automatically connected to the server.

### ► To change what happens when a session ends

- Specify the following parameter in the HTML page:

```
<param name="end" value="manual|auto|terminate|URL">
```

Where:

- **manual** - displays a button when your session ends. Click to reconnect.
- **auto** - displays the **Reconnecting** dialog box when your session ends. The number in the dialog box counts down to 0 and the client reconnects.
- **terminate** - displays either **Connection Terminated** or **Connection Error** when your session ends.
- **URL** - displays the splash screen and redirects to the URL you specified in two seconds. You can specify the URL of any Web page.

The HTML examples supplied with the client include the end parameter and the value specified is a URL to a page called end.html. The applet tag section of each of the examples has this parameter:

```
<param name="End" value="end.html">
```

When you end the session, the client redirects to end.html page which contains a script to close the browser windows. You can edit end.html to display anything you want.

### ► To change the time out period for automatic reconnection

- The default time out period is 5 seconds. To change the period, specify the following parameter in the HTML page:

```
<param name="ReconnectDelay" value="delay">
```

Where *delay* is the delay in seconds. Specifying this parameter does not affect the

delay before connection to a URL if you have specified **URL** for the **end** parameter.

---

**Note** If the start and end parameters are both set to (auto), the startup splash screen is displayed and you must click it to connect.

---

## Keyboard and Mouse

The ICA Java Client lets users specify what type of keyboard to use in their ICA sessions. By default, if you do not specify a keyboard preference, your session uses the keyboard that is specified as the default keyboard for your MetaFrame server.

---

**Note** Different JVMs interpret key detection differently. If you experience problems with unexpected keyboard behaviour, consider upgrading or changing your Web browser.

---

The ICA Java Client supports using any keyboard supported by the MetaFrame server you are connecting to.

### ► To specify a keyboard other than the server's default

- Specify the following parameter in the HTML Page:

```
<param name="user.wfclient.keyboardlayout" value="layout">
```

Where *layout* is a value from the server's list of supported keyboards. Values that contain a space (for example, Swiss German) should be enclosed in quotation marks. Here is a list of the supported keyboards:

```
" (Server Default) "
"Albanian"
"Belarusian"
"Belgian Dutch"
"Belgian French"
"Brazilian (ABNT) "
"British"
"Bulgarian (Latin) "
"Bulgarian"
"Canadian English (Multilingual) "
"Canadian French (Multilingual) "
"Canadian French"
"Croatian"
"Czech (QWERTY) "
"Czech"
"Danish"
"Dutch"
"Estonian"
"Finnish"
"French"
```

---

```
"German (IBM) "  
"German"  
"Greek (220) Latin"  
"Greek (220) "  
"Greek (319) Latin"  
"Greek (319) "  
"Greek Latin"  
"Greek"  
"Hungarian 101-Key"  
"Hungarian"  
"Icelandic"  
"Irish"  
"Italian (142) "  
"Italian"  
"Japanese"  
"Korean"  
"Latin American"  
"Latvian (QWERTY) "  
"Latvian"  
"Lithuanian"  
"Norwegian"  
"Polish (214) "  
"Polish (Programmers) "  
"Portuguese"  
"Romanian"  
"Russian (Typewriter) "  
"Russian"  
"Serbian (Cyrillic) "  
"Serbian (Latin) "  
"Slovak (QWERTY) "  
"Slovak"  
"Slovenian"  
"Spanish Variation"  
"Spanish"  
"Swedish"  
"Swiss French"  
"Swiss German"  
"Taiwan"  
"Turkish (F) "  
"Turkish (Q) "  
"Ukrainian"  
"United Kingdom"  
"US"  
"US-Dvorak for Right hand"  
"US-Dvorak for left hand"  
"US-Dvorak"  
"US-International"
```

For example, to specify a Danish keyboard, create an HTML page like the following:

```
<html>
<body>
<applet code=com.citrix.JICA
archive="JICAEngJ.jar"
width=640 height=480>
  <param name="cabinets" value="JICAEngM.cab">
  <param name="address" value="CitrixServer">
  <param name="user.wfclient.keyboardlayout" value="Danish">
</applet>
</body>
</html>
```

The ICA Java Client supports the following keyboard types to distinguish between subtypes of the Japanese keyboard layout:

```
" (Default) "
" IBM PC/XT or compatible keyboard "
" 101 Keyboard (Japanese) "
" 106 Keyboard (Japanese) "
" NEC PC-9800 on PC98-NX (Japanese) "
" NEC PC-9800 on PC98-NX 2 (Japanese) "
" NEC PC-9800 Windows 95 and 98 (Japanese) "
" NEC PC-9800 Windows NT (Japanese) "
" Japanese Keyboard for 106 (Japanese) "
" DEC LK411-JJ Keyboard (Japanese) "
" DEC LK411-AJ Keyboard (Japanese) "
```

---

**Note** If you are using a 109 key Japanese keyboard, specify the keyboard type “106 Keyboard (Japanese)”.

---

### ► To specify a Japanese keyboard type

- Specify the following parameter in the HTML page:

```
<param name="user.wfclient.keyboardtype"
value="101 Keyboard (Japanese) | 106 Keyboard (Japanese)">
```

For example, to specify a Japanese 106 key keyboard, create an HTML page like the following:

```
<html>
<body>
<applet code=com.citrix.JICA
archive="JICAEngN.jar"
width=640 height=480>
<param name="cabinets" value="JICAEngM.cab">
<param name="address" value="CitrixServer">
```

```

<param name="user.wfclient.keyboardlayout" value="Japanese">
<param name="user.wfclient.keyboardtype"
value="106 Keyboard (Japanese)">
</applet>
</body>
</html>

```

## Using Applications that require a 3-Button Mouse

In the ICA Java Client, the middle button of a 3-button mouse can be emulated by clicking both buttons of a 2-button mouse at the same time.

## Hotkeys

Hotkeys are used to control the behavior of the client and as substitutes for the standard Windows hotkeys for a published application. For example, if you wanted to display the Windows NT Security Desktop on a Windows PC, you would press CTRL+ALT+DEL. If you are running the ICA Client on a Windows PC and are working in a MetaFrame session, this key combination opens the Security Desktop on the local machine. Hotkey functionality allows you to map common key combinations like CTRL+ALT+DEL to a key combination such as CTRL+F1 that will be ignored by your local operating system. When you press this new combination, the ICA Java Client will send CTRL+ALT+DEL to the MetaFrame server, displaying the Windows NT Security Desktop in your session.

The following hotkeys are available:

Hotkey (Default Mapping)	Description
Hotkey2 (SHIFT+F3)	Close Remote Application. The Close Remote Application hotkey closes programs opened in an ICA session. If no programs are open, this hotkey prompts the user that the ICA session will be shut down and then, if given permission, disconnects the ICA session.
Hotkey4 (CTRL+F1)	Substitute for the standard Windows hotkey CTRL+ALT+DEL. The CTRL+ALT+DEL hotkey displays the Windows Security Desktop in the ICA session.
Hotkey5 (CTRL+F2)	Substitute for the standard Windows hotkey CTRL+ESC. On MetaFrame servers, the remote Windows Start menu appears.
Hotkey6 (ALT+F2)	Substitute for the standard Windows hotkey ALT+ESC. This hotkey is used to bring the focus to maximized and minimized windows of programs that have been opened in an ICA session, in the order that they were opened.

Hotkey (Default Mapping)	Description
Hotkey7 (ALT+PLUS)	Substitute for the standard Windows hotkey ALT+TAB. Use this hotkey to cycle through applications that are open in the ICA session. A popup box appears and displays the programs as you cycle through them. The chosen application receives keyboard and mouse focus.
Hotkey8 (ALT+MINUS)	Substitute for the standard Windows hotkey ALT+SHIFT+TAB. Like the ALT+TAB hotkey, this key sequence cycles through applications that are open in the ICA session but in the opposite direction. The chosen application receives keyboard and mouse focus.
Hotkey9 (CTRL+F3)	Substitute for the standard Windows hotkey CTRL+SHIFT+ESC. Use this hotkey to display the task manager.
Hotkey10 (CTRL+F4)	Toggle SpeedScreen Latency Reduction. Toggles mouse click feedback and local text echo on and off.

ICA Java Client hotkeys use a pair of keys. The first key defines the shift state and the second defines the character.

► **To change the hotkey sequence from the default**

- For each hotkey, specify two parameters in the HTML page: one for the shift state and a second for the character state, as follows

```
<param name="user.wfclient.hotkey*shift" value="shiftstate">
<param name="user.wfclient.hotkey*char" value="character">
```

Where \* is the hotkey number, *shiftstate* is the shift state (ctrl, shift, or alt, and *character* is the character. You can choose the character from any of the following:

- f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f12, tab, star, plus, minus, escape or (none)

Specifying (none) for the character disables the hotkey.

The following example describes how to map the Close Remote Application hotkey to the key sequence CTRL+F1 and the ALT+TAB hotkey key sequence to SHIFT+TAB.

```
<html>
<body>
<applet code=com.citrix.JICA
archive="JICAEngN.jar"
width=640 height=480>
<param name="cabinets" value="JICAEngM.cab">
<param name="address" value="CitrixServer">
<param name="user.wfclient.hotkey2shift" value="ctrl">
<param name="user.wfclient.hotkey2char" value="f1">
```

```
<param name="user.wfclient.hotkey7shift" value="shift" >
<param name="user.wfclient.hotkey7char" value="tab" >
</applet>
</body>
</html>
```

## Client Device Mapping

The Citrix ICA Java Client supports client device mapping for connections to MetaFrame servers. *Client device mapping* allows a remote application running on the MetaFrame server to access printers and disk drives attached to the local client computer. The applications and system resources appear to the user at the client workstation as if they are running locally. Ensure that client device mapping is supported on your MetaFrame server before using these features.

---

**Note** Drive mapping, COM port mapping, and client audio are not supported by MetaFrame for UNIX Operating Systems Version 1.0. A hotfix is available for Version 1.1, Feature Release 1, to support client drive mapping.

---

This section includes more information about:

- Mapping client drives
- Mapping client printers
- Mapping COM ports
- Mapping client audio

## Client Drive Mapping

*Client drive mapping* makes any directory mounted on the ICA Client workstation, including CD-ROMs, available to the user during ICA sessions on MetaFrame servers. When a MetaFrame server is configured to allow client drive mapping, users can access their locally stored files, work with them during their ICA sessions, and then save them again either on a local drive or on a drive on the MetaFrame server.

You can only configure drive mapping using the **ICA Java Client Settings** dialog box.

When drive mapping has been configured, the ICA Java Client attempts to use it for all connections. If the server does not support drive mapping, or if the JVM is configured not to allow access to local drives, drive mapping is not available.

**Note** You cannot configure client drive mapping in an HTML page because this is a violation of the client's security. Configure client drive mapping by making the **ICA Java Client Settings** dialog box available to users the first time they log on and instructing them to configure drive mapping. The **ICA Java Client Settings** dialog box is described in the help. See "Copying Client Drive Mapping Configuration" on page 60 for information on how to deploy client drive mapping configuration.

---

## Using Mapped Drives

Once configured, mapped drives are transparent and appear the same as other drives on the server. However, due to the way Java accesses file systems, mapped drives cannot perform the following:

- **Lock files.** Applications sometimes lock the files that are in use to prevent other applications from accessing them at the same time, possibly corrupting them. However, the ICA Java Client cannot lock local files. To prevent file corruption, do not access the same file with two or more applications at the same time.
- **Use file attributes.** Attributes cannot be set on files on a mapped drive.
- **Set date and time.** The date and time stamps are not set on files created or edited on mapped drives.
- **Report drive capacity and usage.** Use the operating system of your local machine to determine the capacity of mapped drives.

## Client Printer Mapping

The ICA Java Client can be configured to map client printers to an ICA session so that client system users can print to their local or network printers. Printers can be created with ICA Client Printer Configuration or by configuring auto-created printers either in the HTML page or in the **ICA Java Client Settings** dialog box. The **ICA Java Client Settings** dialog box is described in the help.

## Auto-Created Printers

You can configure auto-created printers in the HTML page or with the **ICA Java Client Settings** dialog box. Citrix recommends that you use the **ICA Java Client Settings** dialog box for greatest ease of use. When you log on, your auto-created printer appears in Print Manager or in the Printers folder and is ready for use. When you log off, your printer mapping is deleted from the MetaFrame server.



► **To configure auto-created printers**

- Specify the printer name, the port name and the driver, using the following parameters in the HTML page:

```
<param name="user.localclientprinters" value="prntername">
<param name="user.prntername.port" value="portname">
<param name="user.prntername.driver" value="drivername">
```

Where:

- ***prntername*** - Can be any name by which you want to identify the printer.
- ***portname*** - Specify a file name, port name, or printer IP address (or network name and print queue).

When printing to a file, the output file is composed of printer machine code. This file can be sent to a printer using a platform-specific utility. For example, use a command prompt on Windows platforms to send the file to a printer by copying the file to a printer port.

When printing to a port, specify the port. A typical port on Windows is LPT1: On Linux or UNIX systems, the port is similar to /dev/lp0/. Check your operating system documentation for more information.

When printing to a network printer, specify the printer's IP address or network name and print queue (for example, ipaddress:printqueuename or networkname:printqueuename)

The ICA Java Client also allows users to print to standard out or standard error. Specify **stdout** or **stderr** to print to standard out or standard error, respectively.

- ***drivername*** - Specify the printer driver. This name must exactly match the driver name on the Citrix server. Check the driver list on the server. For MetaFrame for Windows servers, double-click **My Computer**, double-click **Printers**, then double-click **Add Printer**. In the Add Printer Wizard, be sure My Computer is selected. Click **Next**. Select a port, such as LPT1:. Click **Next**. The list under Printers contains the printer drivers.

You can also make the printer the default printer for your session by specifying the following parameter:

```
<param name="user.testprinter.comment" value="WFCDefault">
```

Where *testprinter* is the name of the printer. If the MetaFrame server is set to connect only to the user's default printer, this sets your auto-created printer to be the default printer.

---

**Note - Printer Drivers** To auto-create a printer, the MetaFrame server must have the correct printer driver installed, as specified in the **ICA Java Client Settings** dialog, or in the HTML Page using the **Driver** parameter. If the driver is not installed, the printer is not auto-created. In this case, the Citrix administrator must install the correct printer driver on the MetaFrame server.

You can change the list of drivers that appears in the **ICA Java Client Settings** dialog box by editing the ICAPrinters.dat file. See “Changing the List of Printer Drivers that Appears in the ICA Java Client Settings dialog box” on page 61 for more information.

---

### Examples: Configuring Individual Printers

The following example demonstrates auto-creating a printer by specifying parameters on the HTML page. This HTML page is suitable for use only with a Windows client system. In this example, the printer's name is Laser. It is connected to the client's LPT1: port and has a driver named HP LaserJet 4/4M Plus PS.

```
<html>
<body>
<applet code=com.citrix.JICA
archive=JICAEngN.jar
width=640 height=480>
<param name="cabinets" value="JICAEngM.cab">
<param name="address" value="CitrixServer">
<param name="user.localclientprinters" value="Laser">
<param name="user.Laser.port" value="lpt1:">
<param name="user.Laser.driver" value="HP LaserJet 4/4M Plus PS">
</applet>
</body>
</html>
```

In the following example, the printer's name is also Laser and has a driver named HP LaserJet 4/4M Plus PS. The printer is a network printer that exists on a network print server with an IP address of 192.168.1.24 and a print queue named FLOOR2\_LJ

```
<html>
<body>
<applet code=com.citrix.JICA
archive=JICAEngN.jar
width=640 height=480>
<param name="address" value="CitrixServer">
<param name="user.localclientprinters" value="Laser">
<param name="user.Laser.port" value="192.168.1.24:FLOOR2_LJ">
<param name="user.Laser.driver" value="HP LaserJet 4/4M Plus PS">
</applet>
</body>
```

```
</html>
```

### Example: Configuring Multiple Printers

The following example illustrates configuring multiple printers, one of which is a network printer. The port name of a network printer may be a UNC name or the name of a local port that has a network printer mapped to it. In this example, the printer NetPrinter1 is a network printer mapped to the local LPT2 port and it has a driver named HP Laser Jet 400 Series PS.

```
<html>
<body>
<applet code=com.citrix.JICA
width=640 height=480>
<param name="address" value="CitrixServer">
<param name="user.localclientprinters" value="Laser,NetPrinter1">
<param name="user.Laser.port" value="lpt1:">
<param name="user.Laser.driver" value="HP LaserJet 4/4M Plus PS">
<param name="user.NetPrinter1.port" value="lpt2:">
<param name="user.NetPrinter1.driver" value="HP LaserJet 400 Series
PS">
</applet>
</body>
</html>
```

### Printing From MetaFrame for UNIX

Before users can print to a client printer from MetaFrame for UNIX, printing must be enabled by the administrator. For further information about printing from MetaFrame for UNIX Operating Systems, see the *Citrix MetaFrame for UNIX Operating Systems Administrator's Guide*, and the manual pages.

### COM Port Mapping

Client COM port mapping allows devices attached to the client computer's serial ports to be used during an ICA session on the Citrix server. The ICA Java Client supports COM port mapping on the following platforms:

- Windows
- Solaris Sparc

---

**Note** To enable COM port mapping you must have the javax.comm standard extension installed on the client. For Windows and Solaris platforms, you can download the Java Communications API from <http://java.sun.com/products/javacomm>

---

## Mapping a Client COM Port

► **To map a client COM port**

1. Start the ICA Java Client and log on to the MetaFrame server desktop.
2. Start a command prompt: Click **Start**. Click **Programs**, then click **Command Prompt**.
3. At the prompt, type **net use comx: \\client\comz:** where *x* is the number of the COM port on the server and *z* is the number of the COM port you want to map. Press ENTER.
4. To confirm the operation, type **net use** at the prompt. The list that appears contains mapped drives, LPT ports, and mapped COM ports.

---

**Note** COM port mapping is not Microsoft Windows TAPI-compatible. TAPI devices cannot be mapped to client COM ports.

---

---

**Note** On non-Windows systems, the COM ports are numbered in the order they are reported to the ICA Java Client by the Java communications API.

---

## Client Audio Mapping

Client audio mapping enables applications running on the MetaFrame server to play sounds through a sound device installed on the client computer. To use client audio mapping, you must enable it using a parameter in the HTML page.

► **To enable client audio mapping**

- Add the following parameter to the HTML page:

```
<param name="ClientAudio" value="on">
```

You control the amount of bandwidth used by client audio mapping on the MetaFrame server.

► **To change ICA Client audio on a MetaFrame for Windows server**

1. Click **ICA Settings** in Terminal Server Connection Configuration.
2. Select an option from the **Client Audio Quality** drop-down list.

The Client Audio Quality options are:

- 
- **High.** This setting is recommended only for connections where bandwidth is plentiful and sound quality is important. This setting allows clients to play a sound file at its native data rate. Sounds at the highest quality level require about 1.3Mbps of bandwidth to play clearly. Transmitting this amount of data can result in increased CPU utilization and network congestion.
  - **Medium.** This setting is recommended for most LAN-based connections. This setting causes any sounds sent to the client to be compressed to a maximum of 64Kbps. This compression results in a moderate decrease in the quality of the sound played on the client computer. The host CPU utilization will decrease compared with the uncompressed version due to the reduction in the amount of data being sent across the wire.
  - **Low.** This setting is recommended for low-bandwidth connections, including most modem connections. This setting causes any sounds sent to the client to be compressed to a maximum of 16Kbps. This compression results in a significant decrease in the quality of the sound. The CPU requirements and benefits of this setting are similar to those of the Moderate setting; however, the lower data rate allows reasonable performance for a low-bandwidth connection.

---

**Note** If the MetaFrame server is set to use **Low** quality, client audio mapping is disabled.

---

### Audio support on Linux with Java 2 v1.3 platforms

JavaSound, under some circumstances, does not work with the ICA Java Client. Therefore, JavaSound has been disabled for this release on all Linux platforms. Audio will still work in 1.1.x JREs, as these do not use JavaSound.

## Improving Performance

This section describes how you can improve the performance of the ICA Java Client using:

- Data compression
- Data caching
- Queuing mouse movements and keystrokes
- SpeedScreen Latency Reduction
- Improving performance over a low-bandwidth connection

### Data Compression

Data compression reduces the amount of data that needs to be transferred over the network, but requires additional processor resources to compress and decompress the data. If your connection is bandwidth-limited, enabling data compression increases performance. You can specify two levels of data compression, standard and maximum. Using maximum data compression compresses the data further, but uses more processor power and memory and may reduce performance on slow machines. Data compression is enabled by default.

▶ **To enable maximum data compression**

- Specify the following parameters in the HTML page:

```
<param name="maximumcompression" value="on" >
```

▶ **To disable data compression**

- Specify the following parameter in the HTML page:

```
<param name="compress" value="off" >
```

### Bitmap Cache

Bitmap caching stores commonly used graphical objects, such as bitmaps, on a local disk. If your connection is bandwidth-limited, using bitmap caching increases performance. If your client is on a high-speed LAN, you do not need bitmap caching.

Configure disk caching using parameters in the HTML page or with the **ICA Java Client Settings** dialog box.

▶ **To enable disk caching for a connection**

Specify the following parameter in the HTML page:

```
<param name="PersistentCacheEnabled" value="on" >
```

The location of the user.home directory depends on the JVM or Web browser you are using. If necessary, use the **ICA Java Client Settings** dialog box to change the bitmap cache directory.

## Queuing Mouse Movements

When queuing is enabled, the client sends mouse movement updates less frequently to the MetaFrame server. Disabling queuing makes the ICA session more responsive to mouse movements. By default, mouse movement updates are sent every 150 milliseconds.

### ► To change the update period for mouse movement queuing

- Specify the following parameter in the HTML page:

```
<param name="MouseTimer" value="period">
```

Where period is the update period you are using, in milliseconds. To disable mouse movement queuing, set the period to zero.

## Using SpeedScreen Latency Reduction

SpeedScreen Latency Reduction improves performance over high latency connections by providing instant feedback to the user in response to typed data or mouse clicks.

---

**Note** SpeedScreen Latency Reduction works only when it is enabled on the MetaFrame server you are connecting to. See your MetaFrame server documentation for more details.

---

### ► To enable mouse click feedback

- Add the following parameter to the HTML page:

```
<param name="ZLMouseMode" value="0|1|2">
```

Where 0 is off, 1 is on, and 2 is automatic. When mouse click feedback is set to automatic, the client enables it when required.

### ► To enable local text echo

1. Add the following parameter to the HTML page:

```
<param name="ZLKeyboardMode" value="0|1|2">
```

Where 0 is off, 1 is on, and 2 is automatic. When local text echo is set to automatic, the client enables it when required.

## Improving Performance Over a Low-bandwidth Connection

If you are using ICA over a low-bandwidth connection, such as a modem or cellular telephone, there are a number of changes that you can make to your client configuration and the way you use the client to improve performance:

- **Reduce the size of the ICA Java Client applet.** If you do not require the entire functionality of the ICA Java Client, specify only the archives necessary to provide the functions you require. Chapter 2 of this guide explains how to specify component archives. Reducing the size of the applet can greatly reduce the download time.
- **Change your ICA Client configuration.** Changing your client configuration, as described below, can reduce the bandwidth that ICA requires and improve performance.
- **Change the way you use the client.** Changing the way you use the client, as described in “Changing the Way You Use the Client” on page 49, can also reduce the bandwidth required for a high-performance connection.
- **Use the latest MetaFrame server and ICA Clients.** Citrix is continually enhancing and improving ICA performance with each release, and many performance features require the latest client and server software in order to function.

## Changing Your ICA Client Configuration

On devices with limited processing power, or where limited bandwidth is available, there is a trade-off between performance and functionality. The ICA Client provides both user and administrator with the ability to choose an acceptable mixture of rich functionality and interactive performance. Making one or more of the following changes can reduce the bandwidth that your connection requires and improve performance:

- **Enable maximum data compression.** Compression reduces the size of the data that is transferred over the ICA connection.
- **Enable the bitmap cache.** Bitmap caching stores commonly used bitmaps (images) locally on your client so that they do not have to be transferred over the ICA connection every time they are needed.
- **Queue mouse movements.** When queuing is enabled, the client sends mouse updates to the MetaFrame server less frequently. Increasing the queuing period may improve performance on a low bandwidth connection.
- **Enable SpeedScreen Latency Reduction.** SpeedScreen Latency Reduction improves performance over high latency connections by providing instant feedback to the user in response to typed data or mouse clicks.



- **Reduce the window size.** Change the window size to the minimum size you can comfortably use.
- **Reduce the number of colors.** Reduce the number of colors to 256.
- **Disable client audio mapping.** If you do not need sound, disable client audio mapping.

## Changing the Way You Use the Client

ICA technology is highly optimized and typically does not have high CPU and bandwidth requirements. However, if you are using a very low-bandwidth connection, you should consider the following to preserve performance:

- **Avoid accessing large files using client drive mapping.** When you access a large file with client drive mapping, the file is transferred over the ICA connection. On slow connections, this may take a long time.
- **Avoid printing large documents on local client printers.** When you print a document on a local client printer, the print file is transferred over the ICA connection. On slow connections, this may take a long time.
- **Avoid playing multimedia content.** Playing multimedia content uses a lot of bandwidth and can cause reduced performance.

## Integrating the ICA Java Client with Security Solutions

You can integrate the ICA Java Client with a range of security technologies, including proxy servers, firewalls, and SSL/TLS based systems. This section describes:

- Making a connection through a SOCKS proxy server or Secure proxy server (also known as Security proxy server, HTTPS proxy server, or SSL tunneling proxy server)
- Integrating the client with Citrix Secure Gateway or SSL Relay solutions with Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols
- Connecting to a server across a firewall using alternate addressing
- Using ICA encryption to help protect data on local networks

### Connecting Through a Proxy Server

Proxy servers are used to limit access into and out of your network, and to handle connections between ICA Clients and the MetaFrame servers. The ICA Java Client supports both SOCKS and Secure proxy protocols.

### Configuring the ICA Java Client to Use a Proxy Server

To use the ICA Java Client with a proxy server, either enable auto client proxy detection, specify the location of a Proxy Auto Configuration (PAC) file, or manually specify the details of your proxy server in the HTML page or the **ICA Java Client Settings** dialog box.

If you are deploying the ICA Java Client in an organization with many proxy servers, consider using auto client proxy detection. Auto client proxy detection obtains proxy details from the local Web browser settings. It is also useful if you cannot determine which proxy server will be used when you configure the client. Auto client proxy detection can be used with

- Internet Explorer 4.0 or later for Windows
- Netscape 4.x or later for Windows, UNIX, and Linux

You can also automate proxy configuration by using a PAC file. A PAC file is a JavaScript file that is served from a local Web server and used to automatically configure the proxy settings of Web browsers.

If you are manually specifying the proxy server, you will need to know its address. You will also need to know its port number if it is not set to 1080 for a SOCKS proxy server or 8080 for a Secure proxy server.

Some proxy servers require authentication when a connection is requested. Specify a proxy logon name and password in the HTML page when configuring the ICA Java Client. Alternatively, if authentication is required and you do not specify the details in the HTML page, when you open an ICA connection, you are prompted to enter the logon name and password.

## How to Specify a Proxy Server

Configure the ICA Java Client to work with a SOCKS proxy server or Secure Proxy server by specifying parameters in the HTML page, through the **ICA Java Client Settings** dialog box, or through NFuse Classic. This section describes how to specify a proxy server in the HTML page. The **ICA Java Client Settings** dialog box is described in the help. Configuring the ICA Java Client through NFuse Classic is described in the *NFuse Classic Administrator's Guide*.

You can configure the client in one of the following three ways.

▶ **To automatically detect the proxy server settings used by the Web browser (auto client proxy detection)**

- Specify the following parameter in the HTML file:

```
<param name="ProxyType" value="auto">
```

▶ **To obtain the proxy server settings from a PAC file**

- Specify the following parameters in the HTML file:

```
<param name="ProxyType" value="script">  
<param name="ProxyAutoConfigURL"  
value="http://webserver.mycompany.com/myproxies.pac">
```

Where `http://webserver.mycompany.com/myproxies.pac` is the URL for the PAC file.

▶ **To specify the proxy server details manually**

---

**Note** If you are configuring the proxy manually, confirm these details with your security administrator. ICA connections cannot be made if these details are incorrect.

---

- Specify the following details using parameters in the HTML file:
  - the address of the proxy server
  - the port number of the proxy server (if not 1080 for SOCKS or 8080 for Secure proxy)
  - the protocol of the proxy server: SOCKS proxy or Secure proxy
  - in the case of a SOCKS proxy, the protocol version number

- if the proxy requires authentication and you are supplying it through the HTML page, the logon name and password.

Parameter	Description
ProxyType= <i>none</i>   <i>auto</i>   <i>socks</i>   <i>socksv4</i>   <i>socksv5</i>   <i>secure</i>   <i>script</i>	<p><i>none</i> - no proxy</p> <p><i>auto</i> - use the Web browser's settings</p> <p><i>socks</i> - use SOCKS - automatically detect the version</p> <p><i>socksv4</i> - use SOCKS version 4</p> <p><i>socksv5</i> - use SOCKS version 5</p> <p><i>secure</i> - Use Secure Proxy</p> <p><i>script</i> - use a pac file (specified by ProxyAutoConfigURL)</p>
ProxyHost= <i>address:port</i>	Address and port (if required) of the proxy server
ProxyUsername	Proxy Username
ProxyPassword	Proxy Password
ProxyExcludeList= <i>address1; address2; etc...</i>	A semicolon-separated list of address of MetaFrame servers that the client must connect to directly—not via the proxy server.
ProxyAutoConfigURL	The URL for the PAC file. Use with the parameter ProxyType.

For example, to connect to a server named Fountain using a SOCKS proxy server, version 5, at the IP address 10.45.1.3 and port 1080, use an HTML page like the following:

```
<html>
<body>
<applet code=com.citrix.JICA
archive=JICAEngN.jar
width=640 height=480>
  <param name="cabinets" value="JICAEngM.cab">
  <param name="address" value="Fountain">
  <param name="ProxyType" value="socksv5">
  <param name="ProxyHost" value="10.45.1.3:1080">
  <param name="ProxyUsername" value="lornas">
  <param name="ProxyPassword" value="abc123">
</applet>
</body>
</html>
```

---

## Integrating the ICA Java Client with Citrix Secure Gateway or SSL Relay

---

**Note** The SSL and TLS protocols cannot be used with Netscape Navigator 4.x or earlier because the key generation in Netscape Navigator is not sufficiently random for a secure connection.

---

You can integrate the ICA Java Client with a Citrix Secure Gateway or SSL Relay service. The client supports both SSL and TLS protocols:

- SSL provides strong encryption to increase the privacy of your ICA connections and certificate-based server authentication to ensure that the server you are connecting to is a genuine server.
- TLS (Transport Layer Security) is the latest, standardized version of the SSL protocol. The Internet Engineering Taskforce (IETF) renamed it TLS when they took over responsibility for the development of SSL as an open standard. TLS secures data communications by providing server authentication, encryption of the data stream, and message integrity checks. Because there are only minor technical differences between SSL Version 3.0 and TLS Version 1.0, the certificates you use for SSL in your MetaFrame installation will also work with TLS. Some organizations, including US government organizations, require the use of TLS to secure data communications.

For more information on Citrix Secure Gateway, see the *Citrix Secure Gateway Administrator's Guide*.

### Configuring and Enabling the Client for SSL and TLS

SSL and TLS are configured in the same way, use the same certificates, and are enabled with the same parameter. You configure SSL and TLS using parameters in the HTML page or with NFuse Classic.

When SSL and TLS are enabled, each time you initiate a connection the client tries to use TLS first, then tries SSL. If it cannot connect with SSL, the connection fails and an error message appears.

Check that the built-in certificates listed below meet the security requirements for your organization. If your organization has a certificate authority and you need to specify your own certificates, see “Configuring the ICA Java Client for Use With Your Security Solution” on page 55.

Certificate	Issuing Authority
Class4PCA_G2_v2.crt	VeriSign Trust Network
Class3PCA_G2_v2.crt	VeriSign Trust Network
BTCTRoot.crt	Baltimore Cyber Trust Root
GTECTGlobalRoot.crt	GTE Cyber Trust Global Root
GTECTRoot.crt	GTE Cyber Trust Root
Pcs3ss_v4.crt	Class 3 Public Primary Certification Authority
SecureServer.crt	Secure Server Certification Authority

#### ► To enable SSL and TLS

1. If you have not already done so, include the appropriate `cryptoj*.*` and `ssl*.*` archives in the `<archive>` tag in the HTML page. These archives are not included in the complete `JICAEng*.*` archive and you must include them before SSL or TLS encryption can be used.
2. Configure your Web server so that the HTML page specifying the applet can only be delivered to the Web browser via an SSL/TLS (`https://`) connection.

---

**CAUTION** Security is seriously compromised if this step is omitted.

---

3. Change the network protocol to SSL/TLS+HTTPS and enable SSL; add the following parameters to the HTML page:

```
<param name="SSLEnable" value="on">
<param name="BrowserProtocol" value="HTTPTCP">
<param name=HTTPBrowserAddress value=myserver>
```

Where `myserver.mycompany.com` is the fully qualified domain name of a MetaFrame server.

#### ► To force TLS connections

- To force clients to connect only with TLS, you must specify TLS on your Citrix Secure Gateway server or SSL Relay service. See the *Citrix Secure Gateway Administrator's Guide* or SSL Relay service documentation for more information.

## Configuring the ICA Java Client for Use With Your Security Solution

When you enable the SSL/TLS+HTTPS protocol the default settings apply:

- The client attempts to connect using the TLS protocol. If a TLS connection cannot be established, the client tries SSL. If an SSL connection cannot be established, the connection fails and an error message appears.

You must perform further configuration if:

- You are using a Citrix Secure Gateway server which has been configured to run in Relay Mode. If you are not sure what this means, see the Citrix Secure Gateway Administrator's Guide or contact your security administrator for more information .
- You are not using the default Cipher Suite. You may need to use the Com or Gov cipher suites to comply with your organization's security regulations. If you are not sure, contact your security administrator.
- Your organization has its own certificate authority. You may need to specify alternative certificates to comply with your organization's security regulations. If you are not sure, contact your security administrator.

This section describes how to configure these three parameters.

### ► To specify a Citrix Secure Gateway server that has been configured to run in Relay mode

- Add the following parameter to the HTML page:

```
<param name="SSLProxyHost" value="address:port">
```

Where *address* is the fully qualified domain name of the Citrix Secure Gateway and the same domain name specified in the server certificate. If the server port is not 443, specify the port. You can also specify a Citrix Secure Gateway server address in the **ICA Java Client Settings** dialog box.

### ► To Use a Different Cipher Suite

- Add the following parameter to the HTML page:

```
<param name="SSLCiphers" value="All|Com|Gov"
```

Where:

Option	Description
All	SSL_RSA_WITH_RC4_128_MD5 SSL_RSA_WITH_RC4_128_SHA SSL_RSA_WITH_3DES_EDE_CBC_SHA
Com	SSL_RSA_WITH_RC4_128_MD5 SSL_RSA_WITH_RC4_128_SHA
Gov	SSL_RSA_WITH_3DES_EDE_CBC_SHA

### ► To Use Your Own Certificates

You can also use your own SSL certificates with the ICA Java Client. To do this, you need to add your root certificates to an archive, add this archive to the list of Java Client archives, and then specify ICA Java Client parameters to name the root certificates you are using:

- Download a development kit. Before you can create the archive, you must obtain an appropriate kit.
  - To create .jar files for Netscape and other JVMs, download a copy of the Sun JDK from <http://java.sun.com/>
  - To create .cab files for Internet Explorer, download the Microsoft SDK for Java from <http://www.microsoft.com/java/download.htm>
- Create the archive as a .jar or .cab file.
  - To make a .jar file containing the certificates A.crt, B.crt, and C.crt, install the Sun JDK tools (and put them on your path), then type the following at the command line:
 

```
jar -cf MyCertificates.jar A.crt B.crt C.crt
```

 This command generates an archive called MyCertificates.jar. Specify the archive in the HTML page. Note that multiple archives in an applet tag should be separated by commas.
  - To make a .cab file containing the certificates A.crt, B.crt, and C.crt, install the Microsoft SDK for Java (and ensure the CABARC tool is on your path), then type the following at the command line:
 

```
CABARC n MyCertificates.cab A.crt B.crt C.crt
```

 This command generates an archive called MyCertificates.cab. Specify the archive in the cabinets parameter of your HTML tag. Note that multiple archives in a cabinets parameter should be separated by commas.



- Specify which SSL certificates the ICA Java Client should use. Use the following parameters:

**SSLNoCACerts** - The number of specified certificates in the archive. The default value of zero indicates that only the ICA Java Client built-in set of certificates should be used.

**SSLCACert0, SSLCACert1, ... SSLCACert<n>** - The names of SSL root certificates to use for validating the SSL certificate obtained from the SSL Relay. The number of root certificates you specify must match the number specified in the SSLNoCACerts parameter. With these parameters, you can also specify the ICA Java Client built-in certificates, in addition to your own custom certificates (the former reside in the SSL\*. \* archive, the latter reside in the archive you created following the steps above).

Here is an example of an applet tag that enables SSL with the ICA Java Client and uses the custom root certificates A.crt and B.crt, in addition to the ICA Java Client built-in root certificates, Verisign Class 4, and Baltimore.

```
<applet code=com.citrix.JICA width=640 height=480>
  <param name="cabinets"
    value="JICA-coreM.cab,sslM.cab,
    cryptojM.cab,MyCertificates.cab">
  <param name="Address" value="myserver.mycompany.com">
  <param name="SSLEnable" value="on">
  <param name="BrowserProtocol" value="HTTPonTCP">
  <param name="SSLNoCACerts" value="4">
  <param name="SSLCACert0" value="Class4PCA_G2_v2.crt">
  <param name="SSLCACert1" value="BTCTRoot.crt">
  <param name="SSLCACert2" value="A.crt">
  <param name="SSLCACert3" value="B.crt">
</applet>
```

## Connecting to a Server Across a Firewall

Network firewalls can allow or block packets based on the destination address and port. If you are using the ICA Java Client through a network firewall that employs IP address translation, you must specify the following parameters:

- **UseAlternateAddress**: Use an alternate address across a firewall (specified by the parameter HTTPBrowserAddress or TCPBrowserAddress). The values are **0** (default; actual address is used) and **1** (alternate address is used). If this parameter is set to **1**, the parameter HTTPBrowserAddress or TCPBrowserAddress must also be specified. Setting this parameter to **0** is the same as not using the parameter.
- **HTTPBrowserAddress** or **TCPBrowserAddress**: The external Internet address of a MetaFrame server.

**Note** All MetaFrame servers in the farm must be configured with their alternate (external) address. See your MetaFrame server *Administrator's Guide* for more information.

---

For example, to connect to a server across a firewall in applet mode and use an alternate address for the server Fountain, make an HTML page like the following:

```
<html>
<body>
<applet code=com.citrix.JICA
archive="JICAEngN.jar"
width=640 height=480>
<param name="cabinets" value="JICAEngM.cab">
<param name="address" value="Fountain">
<param name="HTTPBrowserAddress" value="177.17.1.7">
<param name="user.wfclient.UseAlternateAddress" value="1">
</applet>
</body>
</html>
```

## Using ICA Encryption

---

**Note** ICA Encryption cannot be used with Netscape Navigator 4.x or earlier because the key generation in Netscape Navigator is insufficiently random for a secure connection.

---

ICA Encryption is included in MetaFrame XP for Windows Feature Release 1 or later. To enable ICA encryption levels higher than **Basic**, you must perform the following tasks:

- Ensure your MetaFrame server is configured to support ICA Encryption.
- If you are using the core archive, include the `cryptoj*.*` and `JICA-sica*.*` archives in the archive attributes or cabinets parameter in the HTML page, if you have not already done so. If you are using a complete archive, only include `cryptoj*.*`.
- Enable strong encryption and set an appropriate encryption level.

### ► To specify ICA encryption

Use the following parameter in an HTML page:

```
<param name="EncryptionLevel" value="0|1|2|5">
```

The number you enter corresponds to the encryption level required as follows. Selecting **Basic** prevents the use of a default encryption level if one has been specified in the past.

Level	Description
0	No encryption (Needs MetaFrame server setup; default is Basic.
1	Basic encryption
2	RC5 128-bit encryption during authentication
5	RC5 128-bit

To create a connection to a MetaFrame server called CitrixServer using 128-bit ICA Encryption using Netscape Navigator 6, create an HTML page like the following:

```
<html>
<body>
<applet code='com.citrix.JICA'
archive="JICA-coreN.jar, cryptojN.jar, JICA-sicaN.jar"
width=640 height=480>
  <param name="cabinets"
    value="JICA-coreM.cab, cryptojM.cab, JICA-sicaM.cab">
  <param name="address" value="CitrixServer">
  <param name="EncryptionLevel" value="5">
</applet>
</body>
</html>
```

## Advanced Configuration

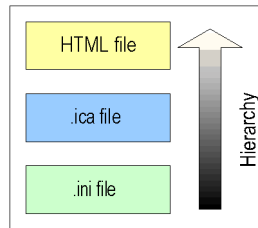
### Precedence of Configuration Methods

This section describes how and where the ICA Java Client stores global preferences and the precedence of each configuration method. It also describes how to deploy configuration settings by copying configuration files.

The ICA Java Client configuration is stored in three places:

- The HTML page, stored on a Web server
- The ICA file, served from an NFuse Classic enabled Web server or standard Web server.
- The appsrv.ini file, stored on your local machine.

Parameters specified in the HTML page take precedence over those specified in the .ica file. Similarly, parameters specified in the ICA file take precedence over those in the appsrv.ini file:



The location of the appsrv.ini configuration file depends on the client operating system and Web browser. The locations for some common Web browser and operating system combinations are as follows:

Sun Plugin 1.3.1 (Win32):

```
%USERPROFILE%\Citrix
```

Microsoft VM (Win32):

```
%SYSTEMROOT%\Java\Citrix
```

Netscape 4 (Win32):

```
<Netscape user directory>\Citrix
```

Netscape 6 (Win32):

```
%USERPROFILE%\Citrix
```

Netscape 4 (UNIX):

```
$HOME/.netscape/.citrix
```

Netscape 6 (UNIX):

```
$HOME/.citrix
```

Internet Explorer (MacOS X):

```
<User's home directory>/ .citrix
```

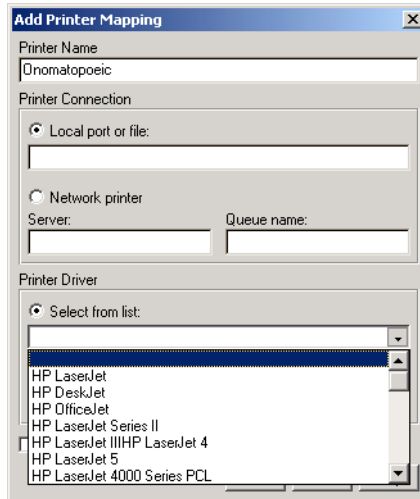
## Copying Client Drive Mapping Configuration

If you are deploying the ICA Java Client to multiple users, you typically configure all common settings through the HTML page, or the ICA file. However, client drive mapping settings can be configured only through the **ICA Java Client Settings** dialog box and are stored only in the appsrv.ini file. They cannot be configured through the .ica file or the HTML page.

You can deploy client drive mapping settings by configuring client drive mapping on one machine and copying the appsrv.ini file to the correct location on the users' machines.

## Changing the List of Printer Drivers that Appears in the ICA Java Client Settings dialog box

When you configure auto-created printers using the **ICA Java Client Settings** dialog box, you set the driver by choosing one from a list:



You can change the list of drivers that appears, to match the printer drivers that are available on your MetaFrame server, by editing the `ICAPrinterDrivers.dat` file. This plain text file is included in the ICA Java Client package and is located in the same directory as the client archives. The contents of the default file are as follows:

```

HP LaserJet
HP DeskJet
HP OfficeJet
HP LaserJet Series II
HP LaserJet IIIHP LaserJet 4
HP LaserJet 5
HP LaserJet 4000 Series PCL
HP LaserJet 4000 Series PS
HP LaserJet 4050 Series PCL
HP LaserJet 4050 Series PS
HP LaserJet 5000 Series PCL
HP LaserJet 5000 Series PS
HP LaserJet 8000 Series PCL
HP LaserJet 8000 Series PS
HP LaserJet 8100 Series PCL
HP LaserJet 8100 Series PS
Canon Bubble-Jet BJC-70
Canon Bubble-Jet BJ-200ex
Canon Bubble-Jet BJC-600
Canon LBP-4

```

```
Canon LBP-8II  
Canon LBP-8III  
Epson Stylus Pro ESC/P 2  
Epson Stylus COLOR ESC/P 2  
Epson Stylus Photo ESC/P 2  
Epson EPL-3000  
Epson EPL-4000  
Epson EPL-5000  
Epson EPL-6000  
Epson EPL-7000  
Epson EPL-8000  
Epson EPL-9000  
Lexmark Optra
```

Use a text editor such as notepad to open the file. Add or remove driver names by deleting or adding names to the file, one driver name per line. You can add the driver names in any order.

## ICA Java Client Parameters

The following is a complete list of the parameters that can be specified in the HTML page to provide additional features and customization. The Address parameter is the only required parameter; all other parameters in these lists are optional.

### General Configuration

Parameter	Description
Address	The address of the MetaFrame server or the name of the published application. If a published application is entered as an address value, the InitialProgram parameter must also be specified.
ICAPortNumber	The default ICA port number is 1494. A different port number can be specified using this parameter or by appending the port number to the address value; for example, "CitrixServer:1495"
InitialProgram	The name of the initial program to run after connecting to the MetaFrame server. If you are connecting to a published application, add a '#' before the program name.
Param	Passes a parameter to a published application such as a filename.
HTTPBrowserAddress	The address of a MetaFrame server. This parameter is used when TCP/IP+HTTP or SSL/TLS+HTTPS browsing has been specified. This parameter is also used to designate groups of primary and backup servers.  Note that when the XML Service on your MetaFrame Server is not configured to use the default port 80, you must append ":<port number>" to this parameter, substituting <port number> with the port number your server's XML Service is configured to use.
TCPBrowserAddress	The address of a MetaFrame server. This parameter is used when your client is on a different subnet than the MetaFrame server. This parameter is also used to designate groups of primary and backup servers. Use with the parameter BrowserProtocol=UDP.
BrowserProtocol	Sets the protocol to use to locate the ICA browser.
WorkDirectory	The path of the working directory where the initial program is run after the user connects to the MetaFrame server.
UseHostname	Specifies to use the hostname as the client name. Use in the form client.wfclient.UseHostname.

<b>Parameter</b>	<b>Description</b>
Clientname	Specifies the client name. Use in the form client.wfclient.Clientname. Use with the parameter UseHostname=off
Username	The user name to use during login.
Password	This is the password of the user. The Password parameter cannot be used to specify an encrypted password. To specify an encrypted password, use an ICA file or .ini file that contains an encrypted password.
Domain	The name of the domain for the user name.
Cabinets	This parameter is used to specify Authenticode signed .cab file archives, used with Internet Explorer for Windows.
Language	The Language parameter causes the ICA Java Client's user interface components to appear in a language other than the language of the client machine.
Icafile	This parameter is used to specify an ICA file for the ICA Java Client to use. The value entered must be a valid URL.
EndSessionTimeout	In Netscape Navigator/Communicator, when users navigate away from a Web page containing an open ICA session for more than 300 seconds (five minutes), the session is disconnected. Use this parameter to specify a different value in seconds.

---



## User Interface Parameters

Parameter	Description
Width	The width of the ICA session window. This parameter is set in the applet tag.
Height	The height of the ICA session window. This parameter is specified in the applet tag.
DesiredColor	The color depth of the ICA session windows. The values are 2 (256 colors), 4 (Thousands of colors), and 8 (Millions of colors). The default value is 256 colors. 16 color mode is not supported but the client can connect to applications published in 16 color mode.
Border	This parameter turns the border around the ICA session in the browser window on or off. Values for this parameter are on or off. The default value is off.
ShowStatusBar	Shows the status bar. You must include the complete archive or the config*. * archive to display the status bar.
ShowSettingsButton	Shows the settings button. You must include the complete archive or the config*. * archive to display the <b>Settings</b> button.
BorderWidth	This parameter allows the user to specify a border width in pixels. The default value is 6.
Start	This parameter controls the ICA Java Client's behavior when you start a session. The values are manual (default) and auto.

End	This parameter controls the ICA Java Client's behavior when you terminate a session. The values are manual (default), auto, terminate, and <i>URL</i> .
KeyboardLayout and KeyboardType	These parameters are used to specify the type of keyboard. Use in the form <code>user.wfclient.KeyboardType</code> and <code>user.wfclient.KeyboardLayout</code> .
Hotkey2Shift Hotkey2Char Hotkey4Shift Hotkey4Char Hotkey5Shift Hotkey5Char Hotkey6Shift Hotkey6Char Hotkey7Shift Hotkey7Char Hotkey8Shift Hotkey8Char Hotkey9Shift Hotkey9Char Hotkey10Shift Hotkey10Char	These parameters set hotkeys that can be used to control various client functions. Use in the form <code>user.wfclient.Hotkey<math>n</math>Shift</code> or <code>user.wfclient.Hotkey<math>n</math>Char</code> where $n$ is the number of the hotkey.

---

## Client Device Mapping Parameters

Parameter	Description
LocalClientPrinters	This parameter is used for passing information about client printers to the server. Must be specified as <param name="user.localclientprinters" value="...">. To specify more than one printer, separate the printer names with commas.
Port	Sets the printer port. Use in the form user. <i>printername</i> .Port where <i>printername</i> is the name allocated with the parameter LocalClientPrinters.
Driver	Sets the printer driver. Use in the form user. <i>printername</i> .Driver where <i>printername</i> is the name allocated with the parameter LocalClientPrinters.
ClientAudio	Use this parameter to enable client audio. The values are on and off. The default is off.

## Performance Tuning Parameters

Parameter	Description
Compress	Sets data compression. The values are 'on' to enable data compression and 'off' to disable it. By default, compression is enabled.
MaximumCompression	Sets high level data compression. The values are 'on' to enable greater data compression and 'off' to select normal data compression. Requires <b>Compress</b> parameter to be enabled.
PersistentCache Enabled	Enables or disables bitmap caching. Must be specified as "user.wfclient.PersistentCacheEnabled".
PersistentCacheSize	Sets the size of the bitmap cache in bytes.
PersistentCacheMin Bitmap	Sets the size of the smallest bitmap to cache, in bytes.
MouseTimer	Sets the period (in milliseconds) between the mouse movement updates that are sent to the MetaFrame server. Set to 0 to disable queuing.
ZLMouseMode	SpeedScreen Latency Reduction mode. The values are 0 (off), 1 (on), or 2 (auto). The default is auto.
ZLKeyboardMode	SpeedScreen Latency Reduction mode. The values are 0 (off), 1 (on), or 2 (auto). The default is off.

## Security Integration Parameters

Parameter	Description
ProxyType	Specifies the type of proxy server.
ProxyHost	Specifies the location and port of the proxy server.
ProxyUsername ProxyPassword	Specifies proxy server login credentials.
ProxyExcludeList	A semicolon or comma separated list of addresses of MetaFrame servers that the client must connect to directly—not via the proxy server.
ProxyAutoConfigURL	Specifies the location of a Proxy Auto Configuration (PAC) file for automatic proxy configuration.
SSLEnable	Enables SSL and TLS encryption protocols. Used with the BrowserProtocol parameter.
SSLProxyHost	Specifies a Citrix Secure Gateway (Relay Mode) address.
SSLCiphers	Specifies an alternative cipher suite.
SSLNoCACerts	Specifies the number of additional certificates.
SSLCACert0... SSLCACert*	Specifies the name of an additional certificate.
UseAlternateAddress	Use an alternate server address across a firewall. Used in the form user.wfclient.UseAlternateAddress.
EncryptionLevel	Sets the level of ICA Encryption to use for an ICA connection.

## Troubleshooting

The following list describes some known problems and suggests a workaround when applicable.

---

**Note** Where appropriate, problems in this list have been reported to the respective institutions.

---

## Server names with non-ASCII characters

Generally, for servers to be contactable by client operating systems, it is necessary for the server's hostname and domain name to be compliant with the Domain Naming System (DNS). That is, containing only upper- and lower-case ASCII a to z, digits 0 to 9 and dash '-'. This may not be necessary when client operating systems and the DNS server support multi-national characters.

Published application names can comprise any characters because these are always resolved by the ICA Browser. Note that any ICA Browser addresses (HTTPBrowserAddress or TCPBrowserAddress) have the same DNS restrictions as server names described above.

On some systems, you may not be able to connect to a server if its name does not conform to this requirement (e.g. a server with a Japanese name or a name containing other non-ASCII characters).

To avoid this, you should either:

- Conform to the DNS requirement and use only ASCII characters in server names.
- or –
- Specify ICA Browser addresses as IP addresses and also do not enable DNS resolution for your Citrix Server. If you do this, then your Citrix Servers can have names that do not comply with the DNS requirement. However, note that SSL will not work in this scenario because DNS names are required for certificate validation, and also NFuse will not work as DNS is used to resolve the hostname part of the NFuse URL.

## Client Drive Mapping

- Remote application generates an error message because it cannot lock a file. Due to Java limitations, Java programs cannot lock files. This means that files cannot be locked on mapped local drives. To avoid this problem, disable the safelock option the HTML page:

```
<param name="safelock" value="off">
```

When safelock is disabled, if a remote application attempts to lock a file on a mapped drive, the ICA Java Client will report success. However, this does not mean that the file has been locked and you must take precautions to ensure that another application does not write to or delete the same file.

## Client Printer Mapping

When using the ICA Java Client on the following Web browser and operating system combinations, the list of printer drivers in the ICA Java Client Settings dialog box does not appear:

- Netscape 4.x on any platform
- Internet Explorer on MacOS 9.x
- Internet Explorer on MacOS X

To do this, you need to add the ICAPrinterDrivers.dat file to an archive and add this archive to the list of Java Client archives specified in the HTML page. The ICAPrinterDrivers.dat file is a plain text file, included in the ICA Java Client package and located in the same directory as the client archives.

### ► To add the ICAPrinterDrivers.dat file to an archive

1. Download the appropriate development kit. Before you can create the archive, you must obtain an appropriate kit.

- To create .jar files for Netscape and other JVMs, download a copy of the Sun JDK from <http://java.sun.com/>
- To create .cab files for Internet Explorer, download the Microsoft SDK for Java from <http://www.microsoft.com/java/download.htm>

2. Create the archive as a .jar or .cab file.

- To make a .jar file containing ICAPrinterDrivers.dat, install the Sun JDK tools (and put them on your path), then type the following at the command line:

```
jar -cf PrinterDrivers.jar ICAPrinterDrivers.dat
```

This command generates an archive called PrinterDrivers.jar. Specify the archive in the HTML page. Note that multiple archives in an applet tag should be separated by commas.

- To make a .cab file containing ICAPrinterDrivers.dat, install the Microsoft SDK for Java (and ensure the CABARC tool is on your path), then type the following at the command line:

```
CABARC n PrinterDrivers.cab ICAPrinterDrivers.dat
```

## Client Audio Mapping

- Client audio mapping may not work on Netscape 4.x running on Linux. Audio mapping appears to function on the MetaFrame server, but no sound is output on the client.

- Starting the ICA Java Client while an audio application is running on your desktop can disable audio mapping. Do not run audio applications while starting the ICA Java Client.

## Clipboard support on X11

- ICA Java Client clipboard support only works with applications that use the X11 CLIPBOARD selection. Specifically, Motif and Gnome applications are fine, but KDE applications do not work. Xterm can be configured with X11 resources to use the CLIPBOARD selection, for example:

```
XTerm*VT100.Translations: #override \n\  
~Ctrl ~Meta <Btn2Up>:\n\  
insert-selection(PRIMARY,CLIPBOARD,CUT_BUFFER0)\n\  
<BtnUp>: select-end(PRIMARY,CLIPBOARD,CUT_BUFFER0)\n
```

GNU Emacs can also be configured to use CLIPBOARD with the following Emacs lisp:

```
(setq x-select-enable-clipboard t)
```

- Also, the xclipboard utility (present on most X11 systems) may be of use in transferring data between the PRIMARY and CLIPBOARD selections.

## Mouse Pointer Support

Java 1.1 only supports 14 different mouse pointers, so in these environments the Java Client makes a best effort attempt to map remote pointer shapes to one of these 14. Java 2 supports arbitrary mouse pointers, so in these environments the ICA Java Client will be able to display a correct mouse pointer. You can obtain a Java 2 applet environment by installing Sun's Java Plug-in.

## Internet Explorer

- When you click the **Back** button, the applet stops. If you revisit the page, the applet restarts.

To navigate to other pages, open a new browser window.

---

**Note** Other browsers exhibit similar behavior under certain circumstances. For example, Netscape Navigator allows a limited number of applets to run. If you navigate away from your ICA session and run several applets, when you navigate back your ICA session may restart.

---

- The browser does not correctly route some ALT key events. For example, pressing ALT+F when an application running in the ICA session has focus sends the event to both the application and to the browser.

- When using scroll bars images sometimes repaint incorrectly.
- To display dead key characters, you must press the chosen dead key and then press the space bar twice.
- Using Microsoft's JVM in Internet Explorer, buttons on dialog boxes do not automatically have keyboard focus.

## Netscape Navigator

- The ICA Java Client does not support ICA Encryption or SSL/TLS protocols on Netscape Navigator 4.x. If you require encryption, upgrade to Netscape 6.x.
- Netscape Navigator/Communicator 4.x for MacOS 9 does not fully support 1.1 applets. Netscape Navigator/Communicator 4.76 and later using the Java Plugin (MRJ Plugin) does work; it has not been extensively tested.
- The Java Environment under Netscape 4.x does not correctly report the system's locale and the Java Client applet always displays English text. To work around this problem, you need to explicitly specify the language as an applet parameter, for example:

```
<applet code=com.citrix.JICA archive="JICAEngN.jar" width=640
height=480>
<param name="Language" value="es">
</applet>
```

- Auto-locate with UDP broadcast is not supported in the Netscape 4.x JVM. To work around this limitation, specify a server location. See “Configuring Network Protocol and Server Location” on page 28.

## MacOS 9.x

- Internet Explorer 4.0 on Macintosh client systems does not fully support signed .cab files. Use the \*.jar files instead.
- When using Internet Explorer on Macintosh client systems, the TAB key cannot be used in a MetaFrame session. This is because Internet Explorer uses the TAB key to move keyboard focus within the Web browser, and consequently does not get sent to the MetaFrame server. This is a limitation of Internet Explorer.
- CTRL keys on Macintosh cannot be correctly detected.
- To right-click when connected to a Citrix server with a Macintosh, hold down the command key and click the mouse button.
- To display dead key characters on Macintosh systems, you must press the chosen dead key and then press the space bar twice.



- Characters generated using the Option keys on Macintosh may not be supported by the current Windows font in your ICA session. If the character produced is not the expected character, choose a Windows font in your ICA session that supports the character. After producing the desired character, you can return to your usual font.

## Miscellaneous

- On MacOS 9.x and MacOS X, the help window for the **ICA Java Client Settings** dialog box does not respond to the mouse and keyboard. To work around this issue, close the Settings dialog box before using the help window.
- When connecting to a MetaFrame server, the ICA Java Client cannot detect the state of the Scroll Lock, Number Lock, or Caps Lock keys when the client receives focus.
- Java 2, Standard Edition Version 1.2 is not recommended for use with the ICA Java Client. Upgrade to a Web browser with Java 2, Standard Edition Version 1.3 or upgrade the JVM in your Web browser to Java 2, Standard Edition Version 1.3.
- Persistent caching for high resolution or color depth is only available with MetaFrame XP 1.0 Feature Release 1 or later and MetaFrame for UNIX 1.0 Feature Release 1 or later.
- In rare circumstances, typing at the keyboard can produce no effect. If this happens, setting your Keyboard Layout explicitly should resolve the problem. See “Keyboard and Mouse” on page 34.
- SSL/TLS is not supported when using key lengths greater than 2048 bits.



# Index

## A

- Alternate address translation 57
- Archive tag
  - Syntax 21
- ASCII characters, in server names 69
- Auto-client proxy detection 51

## B

- Before you Begin 7
- Bitmap Cache 46
- Business recovery 28

## C

- Certificates
  - Built in 53
  - Specifying your own 56
- Characters, in server names 69
- Cipher Suite
  - Client configuration 55
- Citrix Secure Gateway
  - Relay mode 55
- Citrix Secure Gateway - client configuration 53
- Citrix Web site 9
  - Frequently Asked Questions 10
  - Product documentation 9
- Client Drive Mapping 39
- Codebase
  - Definition 21
- Codebase parameter 20
- Colors
  - Changing the window properties 32
- COM port mapping 43
- Conventions in the manual 8
- Creating a deployment Web page 20

## D

- Data Compression 46
- DNS resolution 69
- Drive Mapping 39

## E

- Enable bitmap caching 46
- Encryption 50, 53

- Euro-currency support 39

## F

- Firewalls
  - Client configuration 57

## H

- Hotkeys
  - Configuring 37
- HTML page setup for ICA Java Client 20

## I

- ICA Clients
  - downloading from the Citrix Web site 9
- ICA Encryption
  - Client configuration 58
- ICA Java Client Settings dialog box 32
- Introduction 14
- Introduction to the ICA Java Client 11

## J

- JavaSound 45

## K

- Keyboards
  - Types 34

## L

- Linux 45
- Local text echo 47
- Locale
  - Specifying 27

## M

- Mapped Drives
  - Using 40
- Mapping Client Devices 39
- Mouse-click feedback 47

## N

- Network Protocol 28

**P**

- PAC files
  - Proxy server configuration 51
- Parameter Passing 31
- Parameters
  - Complete list 63
  - Syntax 21
- Performance
  - Improving 48
- Printing
  - Auto-created printers 40
  - MetaFrame for UNIX 43
- Proxy server
  - Manual configuration 51
- Proxy servers
  - client configuration 50
  - Secure 50
  - SOCKS 50
  - supported types 50

**Q**

- Queuing mouse movements 47

**R**

- Relay mode 55
- Requirements 15

**S**

- Secure proxy 50
- Security
  - Citrix Secure Gateway 53
  - Firewalls 57
  - ICA Encryption 58
  - integrating the client 50
  - Proxy servers 50
  - SSL Relay 53
- Server Location 28
- Server names 69
- Settings button
  - Displaying 32
- SOCKS proxy 50
- SpeedScreen Latency Reduction 47, 67
- SSL
  - client configuration 53
- Starting the ICA Java Client
  - Specifying locale 27

- Status Bar
  - Displaying 32
- System Requirements 15

**T**

- TCP/IP server location 28
- Three-button mouse
  - Support 37
- TLS
  - Client configuration 53
- Transport Layer Security 53
- Troubleshooting 68
  - Internet Explorer 4.0 71
  - Miscellaneous 73
  - Netscape 72

**W**

- Web page
  - Deploying the client 20