Citrix ICA Client Simulation
API Specification

Programmer's Guide

MetaFrame Presentation Server Client for 32-bit Windows, Version 9.*x*

Citrix® MetaFrame® Presentation Server 4.0 for Windows®
Citrix® MetaFrame® Access Suite

Last Updated: December 17, 2004 2:33 pm (KT)

Contents

Welcome

The ICA Simulation API extends the ICA Client Object specification and provides scripting interfaces that enable MetaFrame Presentation Server Clients to operate in a non-user-interface or "headless" mode. Citrix recommends that you use this guide in conjunction with the ICO Client Object Programmer's Guide which is available for download from the Citrix Developer Network (CDN).

Who Should Read this Book?

This guide is for Citrix server administrators, ISVs, and power users of the ICA Client who need to monitor and optimize load testing and performance management aspects of Citrix-based application environments. This guide assumes knowledge of:

- MetaFrame Presentation Server for Windows with Feature Release 2 or later
- ICA Client for 32-bit Windows, Version 8.x or later
- Citrix ICA Client Object

Use this guide in conjunction with:

- Citrix ICA Client Object Programmer's Guide
- MetaFrame Presentation Server Administrator's Guide
- MetaFrame Presentation Server Client Administrator's Guides

It is assumed that you have the ICA Client for 32-bit Windows (Version 8.x or later) installed and working on the client device, and are familiar with basic embedding and scripting tools and techniques.

Components of the ICA Simulation API Distribution Kit

The ICA Simulation API distribution kit includes the following components:

- **Programmer's Guide.** This guide, which includes high level information about the ICA Simulation object model and an API reference that describes all objects exposed by the ICA Simulation API.
- **Sample code.** Two sample programs that represent typical application of the ICA Simulation API. For more information, see "Sample Programs" on page 16.

Hardware, Software, and Developer Tool Requirements

Because you will use the ICA Simulation API to develop applications for a MetaFrame Presentation Server environment, you should have the hardware and software needed to support MetaFrame Presentation Server Feature Release 2 or later. For hardware and software requirements and for installation procedures, see the *MetaFrame Presentation Server Administrator's Guide*.

To create a test application using the ICA Simulation API, you can use any COM-based programming language; the ICA Simulation distribution kit includes sample programs written using Visual Basic and Visual C++.

Execution Environment

The ICA Simulation API can be used only to develop programs that will access servers running MetaFrame Presentation Server Feature Release 2 or later. Previous versions of MetaFrame Presentation Server are not supported; programs that use the ICA Simulation API require the ICA Win32 Client, Version 7.0 or later.

Getting More Information and Help

This section describes how to get more information about MetaFrame Presentation Server and how to contact Citrix.

Accessing Product Documentation

The documentation for MetaFrame Presentation Server includes online documentation, known issues information, integrated on-screen assistance, and application help.

- Online documentation is provided in Adobe Portable Document Format (PDF) files. Online guides are provided that correspond to different features of MetaFrame Presentation Server. For example, information about the Web Interface is contained in the *Web Interface Administrator's Guide*. Use the *Document Center* to access the complete set of online guides.
- In many places in the MetaFrame Presentation Server user interface, integrated on-screen assistance is available to help you complete tasks. For example, in the Access Suite Console, you can position your mouse over a setting to display help text that explains how to use that control.
- Online help is available in many components. You can access the online help from the Help menu or Help button.

Important To view, search, and print the PDF documentation, you need to have Adobe Reader 5.0.5 with Search or a later version with Search. You can download Adobe Reader for free from the Adobe Web site at http://www.adobe.com/.

Document Conventions

MetaFrame Presentation Server documentation uses the following typographic conventions for menus, commands, keyboard keys, and items in the program interface:

Convention	Meaning
Boldface	Commands, names of interface items such as text boxes and option buttons, menu and tab names, and user input.
Italics	Placeholders for information or parameters that you provide. For example, *filename* in a procedure means you type the actual name of a file. Italics also are used for new terms and the titles of books.
UPPERCASE	Keyboard keys, such as CTRL for the Control key and F2 for the function key that is labeled F2.

Convention	Meaning
`Monospace`	Text displayed in a text file.
%SystemRoot%	The Windows system directory, which can be WTSRV, WINNT, WINDOWS, or other name specified when Windows is installed.
{ braces }	A series of items, one of which is required in command statements. For example, **{ yes | no }** means you must type **yes** or **no**. Do not type the braces themselves.
[brackets]	Optional items in command statements. For example, [**/ping**] means that you can type **/ping** with the command. Do not type the brackets themselves.
| (vertical bar)	A separator between items in braces or brackets in command statements. For example, **{ /hold | /release | /delete }** means you type **/hold** or **/release** or **/delete**.
… (ellipsis)	You can repeat the previous item or items in command statements. For example, **/route:***devicename*[,…] means you can type additional *devicenames* separated by commas.

Getting Service and Support

Citrix provides technical support primarily through the Citrix Solutions Network (CSN). Our CSN partners are trained and authorized to provide a high level of support to our customers. Contact your supplier for first-line support or check for your nearest CSN partner at http://www.citrix.com/support/.

In addition to the CSN channel program, Citrix offers a variety of self-service, Web-based technical support tools that include the following:

- The Citrix Knowledge Center, an interactive tool containing thousands of technical solutions to support your Citrix environment
- Support Forums, where you can participate in technical discussions and search for previous responses from other forum members
- Access to the latest service packs, hotfixes, and utilities
- Downloadable clients, available at http://www.citrix.com/download/

Another source of support, Citrix Preferred Support Services, provides a range of options that allows you to customize the level and type of support for your organization's Citrix products.

Subscription Advantage

Subscription Advantage gives you an easy way to stay current with the latest server-based software functionality and information. Not only will you get automatic delivery of feature releases, software upgrades, enhancements, and maintenance releases that become available during the term of your subscription, you also get priority access to important Citrix technology information.

You can find more information on the Citrix Web site at http://www.citrix.com/services/ (select Subscription Advantage). You can also contact your Citrix sales representative or a member of the Citrix Solutions Network for more information.

Customizing MetaFrame Presentation Server

The Citrix Developer Network (CDN) is at http://www.citrix.com/cdn/. This open-enrollment membership program provides access to developer toolkits, technical information, and test programs for software and hardware vendors, system integrators, ICA licensees, and corporate IT developers who incorporate Citrix computing solutions into their products.

Most of the operations that you can perform using the MetaFrame Presentation Server user interface can also be scripted by using the Citrix Software Development Kit (SDK). The SDK also lets programmers customize most aspects of MetaFrame Presentation Server. The Citrix Server Software Development Kit (SDK) is available from http://www.citrix.com/cdn/.

Education and Training

Citrix offers a variety of instructor-led training and Web-based training solutions. Instructor led courses are offered through Citrix Authorized Learning Centers (CALCs). CALCs provide high quality classroom learning using professional courseware developed by Citrix. Many of these courses lead to certification.

Web-based training courses are available through CALCs, resellers, and from the Citrix Web site.

Information about programs and courseware for Citrix training and certification is available from http://www.citrix.com/edu/.

Disclaimer

This documentation is provided to you, the licensed user, by Citrix Systems, Inc. ("Citrix") pursuant to the terms of the Software Development Kit License Agreement included as Appendix A and may only be used in accordance with the express terms of the Software Development Kit License Agreement.

This document is provided "AS IS" and Citrix and its suppliers make and the licensed user receives no warranties or conditions, express, implied, statutory, or otherwise. Citrix has made reasonable efforts to ensure the completeness and accuracy of all information contained in this document.

This document is subject to change without notice and Citrix is not obligated to maintain, update, or otherwise support such documentation.

ICA Simulation API Specification

Purpose

MetaFrame Presentation Server is the industry-leading application serving platform providing unparalleled server-based application management, device access, and security. Citrix solutions are tailored to provide remote office connectivity, workforce mobility, business continuity, and centralized application deployment. A Citrix application-serving environment is efficient, flexible, and cost-effective for application delivery and administration and is now being rapidly adopted by the corporate mainstream.

With Citrix application-serving environments now scaling to hundreds of servers, and tens of thousands of users, it is becoming imperative to perform effective performance testing and monitoring of server-based applications. This is essential for test, as well as production environments.

Testing and monitoring applications enable Citrix customers to perform accurate capacity planning or sizing. With thorough testing, Citrix customers can obtain accurate performance information to determine the number of server farms required, whether to scale horizontally with large numbers of small servers, or to scale vertically with fewer but larger systems. This information allows companies to deploy environments with the best price-to-performance ratios.

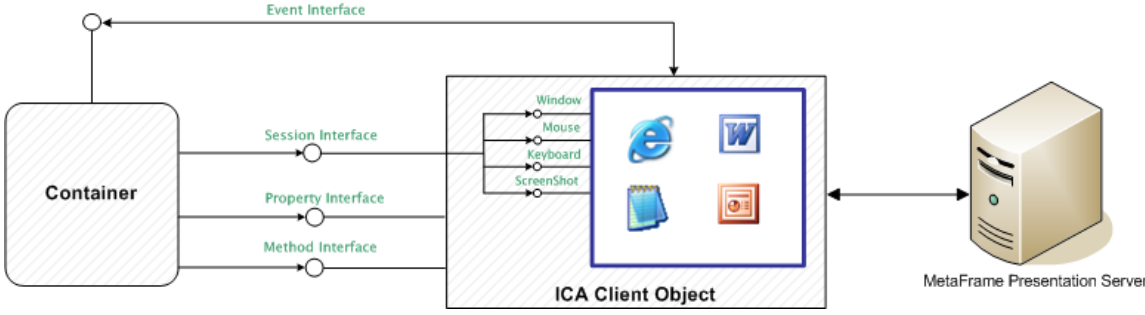
The ICA Simulation API Specification is primarily targeted at independent software vendors (ISVs) who are interested in providing native support for the Independent Computing Architecture (ICA) protocol through their respective enterprise testing solutions.

The aim is to encourage the development of robust, industry-standard, enterprise test solutions that can emulate, test, tune, and monitor Citrix-based application serving environments. By using commercial enterprise testing solutions, Citrix customers can derive greater benefit from MetaFrame Presentation Server products, as well as applications accessed through MetaFrame Presentation Server Clients, allowing them to deliver optimal performance to all users.

Unlike the Citrix Server Test Kit, which is used to develop server-based testing mechanisms, you can use the ICA Simulation API to develop test solutions that can interact with the ICA protocol on the client device, thereby providing an end-to-end realistic environment for testing and tuning activities.

The ICA Simulation API and the ICA Client Object

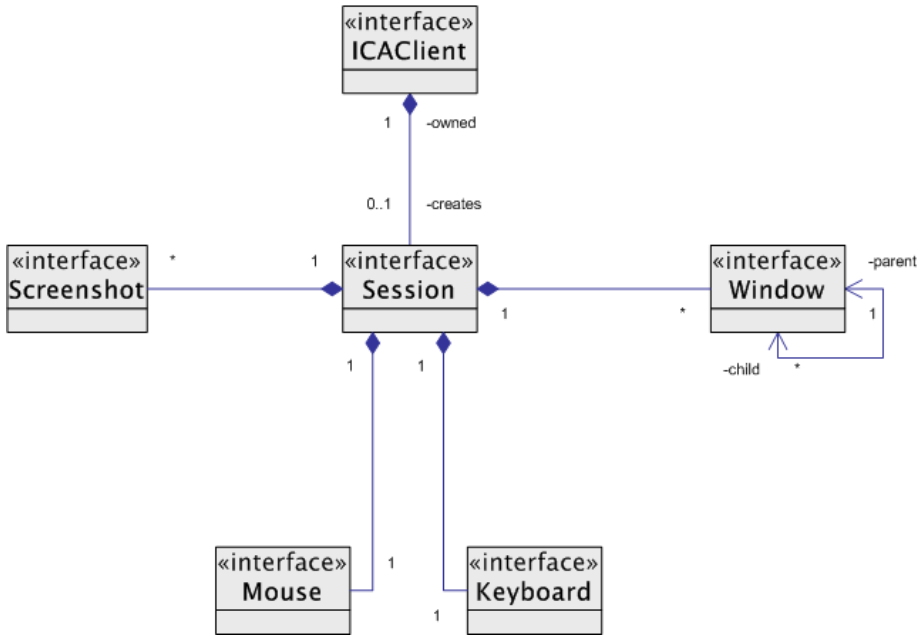
The ICA Client Object (ICO) is the framework that exposes a public interface to the ICA Win32 Client. The ICA Client Object provides a set of standard scripting interfaces or APIs that enable a third-party application to interface and control the ICA Client.



The ICA Simulation API specification extends the ICO specification by providing additional scripting interfaces that can be used by COM-based applications to force the ICA Client to operate in non-user-interface or "headless" mode.

The ICA Simulation Object Model

ICA Simulation API defines a scriptable object model that enables external control and monitoring of certain attributes of a client session. A diagrammatic representation of the ICA Simulation interface is shown below.



As shown in the diagram above, a single ICO object creates a single session object instance. The session object exposes five internal COM objects:

- **Session** represents the interface to a connected ICA session
- **Window** represents the interface to an ICA session window
- **Mouse** represents the interface to the mouse (pointing device) associated with the session object
- **Keyboard** represents the interface to the keyboard (input device) associated with the session object
- **ScreenShot** represents the interface to the current image in the active ICA session window.

These objects are all associated, directly or indirectly, to the session object and cannot be created unless the container first obtains a session object.

ICO Events

To enable scripting languages to use the ICA Simulation object model, all new events defined for window, mouse, keyboard, and session are also delivered to the ICO object.

The ICA Simulation API is designed to be used in compiled C++ or Visual Basic container applications. Though the methods and properties of the available objects can be used in scripts, event notification support is very limited.

Events related to the window object help identify the application window involved. However, an identifying parameter is not required by the session, mouse, keyboard and screenshot objects, which are independent and accessible from the ICA Client Object.

Data Types

The following data types are used by properties and methods available in the ICA Simulation API. Only commonly used data types are described here; data types defined for a specific ICA Simulation function are described in the function description.

Data Type	C++	Visual Basic
Integer	LONG	Long
Boolean	VARIANT_BOOL	Boolean
Enumeration	IWindows (IEnumVARIANT)	IWindows (Collection)
String	BSTR	String

Usage Examples

The following are brief JavaScript examples. Programming languages such as Visual Basic and Visual C++ provide equivalent access mechanisms.

Obtain Session Object

```
var client = new ActiveXObject(“Citrix.ICAClient”)
client.Connect
...
function client_OnConnect() {
client.session.keyboad.SendKeyDown( 145, 0 );
}
```

List Windows

```
for ( e = new Enumerator(client.session.topLevelWindows) ;!e.atEnd();e.moveNext()) {
    var window = e.item();
    document.writeln( “Name: ”, window.name, “ Style: ”,
window.style);
}
```

Ping Server

```
client.session.sendPingRequest(“A String”));
function client_OnPingAck( s, t ) {
  document.writeln( “reply received in ”, t, “ms”);
}
```

Sample Programs

The ICA Simulation API distribution kit ships with two sample programs that represent typical applications of the ICA Simulation API. These are:

ATLico

This is an ATL-based application written using Visual C++ 6. ATLico demonstrates an automated record and replay application that uses the following ICA Simulation API functionality:

- Accessing the ICA Client object model
- Receiving keyboard and mouse events
- Replaying events from a file

ATLico was intentionally written to be simple and compact. The files EventRecorder.cpp and EventRecorder.h contain the single class implementation and definition, respectively, that implements all the application functionality. This class is both a receiver of all ICA Client events and a UI (using CDialog) class containing all the controls. While ATLico encapsulates the record and replay data to a text file, a real recorder application decomposes record and replay data into a number of classes.

ATLico.dsw is a Visual Studio 6.0 workspace and ATLico.dsp is its contained project.

VBwindows

This is a Visual Basic 6 program that monitors and interacts programmatically with an ICA session. This program demonstrates the following ICA Simulation API functionality:

- Accessing the ICA Client Object model
- Receiving keyboard and mouse events
- Interacting with windows within the ICA session
- Using the PING functionality exposed by the ICA Simulation API

VBwindows is written using Visual Basic 6 and was intentionally designed to be simple and compact. The file, window.cls, contains the single class implementation that implements the IWindow interface defined in the ICA Simulation API.

Project.vbw is a Visual Basic 6.0 workspace and Project.vbp its contained project.

Functions

The following sections describe the ICA Simulation Application Programming Interface (API) for the Win32 Client, Wfica32.exe, running in non-user-interface or headless mode.

Important The return values for function prototypes described in this document apply only to C++. In scripting environments, these return values are interpreted by the scripting engine.

Session

Session represents an ICA session and provides the following functionality:

- Enables inspection of mouse and keyboard information and control of client operating in non-user interface mode
- Enables maintenance of a list of currently open top level windows and the window currently in the foreground of the ICA session
- Notifies container about window creation or destruction, keyboard, and mouse events
- Includes a ping function that sends a round-trip message to the Citrix server and notifies the container when the ping completes

Properties

Property	Type	Access	Description
ForegroundWindow	Window	R	Specifies the window currently in the foreground within the ICA session. NULL if a session does not exist or if the window currently in the foreground is not in the list of windows on the client (for example, when the server session desktop window details are not sent to the client).
TopLevelWindows	Enumeration	R	Lists the top level window objects open within this session. Returns an empty list if a session does not exist.
Mouse	Mouse	R	Gets the mouse object.
Keyboard	Keyboard	R	Gets the keyboard object.
ReplayMode	Boolean	RW	TRUE, FALSE Set to TRUE to prevent hardware-generated mouse and keyboard events on the client device being accepted by the ICA session.

Methods

Method	Calling Convention	Description
CreateFullScreenShot	HRESULT CreateFullScreenShot([out, retval] IScreenShot * *pVal)	Creates an object representing an image of the entire ICA session screen (fails with E_FAIL if ICA Simulation mode is RenderLess).
CreateScreenShot	HRESULT CreateScreenShot(Integer x, Integer y, Integer width, Integer height, IScreenShot **pVal)	Creates an object representing the image of the specified area of the ICA session screen (fails with E_FAIL if ICA Simulation mode is RenderLess).
SendPingRequest	HRESULT SendPingRequest(String pingInfo)	Sends a ping synchronization request to the server, with PingInfo as the payload. This triggers the OnPingAck event when the ACK is received.

Events

Event	Calling Convention	Description
OnWindowCreate	HRESULT OnWindowCreate (Window window)	Triggers when a new window is created within the session.
OnWindowDestroy	HRESULT OnWindowDestroy (Window window)	Triggers when an existing window is destroyed within the session. The window cannot be modified once destroyed.
OnPingAck	HRESULT OnPingAck (String pingInfo, Integer roundTripTime)	Indicates that a ping acknowledge message was received and returns the given PingInfo string and the roundTripTime (in milliseconds).
OnWindowForeground	HRESULT OnWindowForeground (Long WindowID)	Notifies the container that a window is being brought into the foreground. Due to the asynchronous nature of the ICA client-server communications, it is possible for the client to receive a foreground notification for a window even before the window creation notification is received. By informing the container about the window in the foreground, the container can cache this ID if the window hasn't been created. After the OnWindowCreate( ) notification is received, the container can use the cached ID to infer if this window is in the foreground or not. If a window exists and is brought into the foreground, OnActivate and OnWindowForeground are triggered.

Usage Guidelines

The session object is exposed by the ICA Client Object (ICO). A session object can be obtained only if an active connection to MetaFrame Presentation Server exists. You can obtain interfaces to keyboard and mouse input streams flowing through a connected session through the session object.

The keyboard stream allows sending of key presses (SendKeyUp) and releases (SendKeyDown) using standard key definitions. The mouse stream allows sending of mouse events using SendMouseUp, SendMouseDown, and SendMouseMove.

To avoid mixing recorded keyboard and mouse events with client-side console keyboard and mouse events, set ICO to replay mode. You can do this by setting ReplayMode to TRUE.

Note If ReplayMode property is set to FALSE, the replay session may produce unexpected results.

Window

Exposes features of an underlying OS window on the server. Server OS windows appearing in the client list must be visible. In this model, windows are created and destroyed to reflect changes to windows on the server.

Top Level Window Properties

An OS window is listed in the TopLevelWindows property of the associated session.

Property	Type	Access	Description
PositionX	Integer	R	Specifies the X-coordinates of the window
PositionY	Integer	R	Specifies the Y-coordinates of the window
Width	Integer	R	Specifies the width of the window
Height	Integer	R	Specifies the height of the window
Style	Integer	R	Specifies the Win32 window style bits
ExtendedStyle	Integer	R	Specifies the Win32 window extended style bits
Caption	String	R	Specifies the caption of the window
LargeIconHash	String	R	Specifies a hash total computed from the application icon's (ALT-TAB) pixel values.
SmallIconHash	String	R	Specifies a hash total computed from the application icon's (task bar) pixel values.
Disposed	Boolean	R	Listener to session events receives TRUE if this window is no longer active (OnWindowDestroy).
WindowFlags	Integer	R	Specifies the ICA window style bits
ParentID	Integer	R	Specifies a unique Identifier for the parent window
WindowID	Integer	R	Specifies a unique Identifier for this window

Window Style Values

Window Style is an ORed combination of the following bitmasks. These values match those defined for Win32 window styles (WS_xxxx macros in Winuser.h for C programmers).

WindowStyle	Value
WindowStyleOVERLAPPED	0x00000000UL
WindowStylePOPUP	0x80000000UL
WindowStyleCHILD	0x40000000UL
WindowStyleMINIMIZE	0x20000000UL
WindowStyleVISIBLE	0x10000000UL
WindowStyleDISABLED	0x08000000UL
WindowStyleCLIPSIBLINGS	0x04000000UL
WindowStyleCLIPCHILDREN	0x02000000UL
WindowStyleMAXIMIZE	0x01000000UL
WindowStyleCAPTION	0x00C00000UL
WindowStyleBORDER	0x00800000UL
WindowStyleDLGFRAME	0x00400000UL
WindowStyleVSCROLL	0x00200000UL
WindowStyleHSCROLL	0x00100000UL
WindowStyleSYSMENU	0x00080000UL
WindowStyleTHICKFRAME	0x00040000UL
WindowStyleGROUP	0x00020000UL
WindowStyleTABSTOP	0x00010000UL
WindowStyleMINIMIZEBOX	0x00020000UL
WindowStyleMAXIMIZEBOX	0x00010000UL

Extended Window Style Values

Extended Window Style is an ORed combination of the following bitmasks. These values match those defined for Win32 extended window styles (ES_xxxx macros in Winuser.h for C programmers).

WindowsExStyle	Value
WindowsExStyleDLGMODALFRAME	0x00000001UL
WindowsExStyleNOPARENTNOTIFY	0x00000004UL
WindowsExStyleTOPMOST	0x00000008UL
WindowsExStyleACCEPTFILES	0x00000010UL
WindowsExStyleTRANSPARENT	0x00000020UL
WindowsExStyleMDICHILD	0x00000040UL
WindowsExStyleTOOLWINDOW	0x00000080UL
WindowsExStyleWINDOWEDGE	0x00000100UL
WindowsExStyleCLIENTEDGE	0x00000200UL
WindowsExStyleCONTEXTHELP	0x00000400UL
WindowsExStyleRIGHT	0x00001000UL
WindowsExStyleLEFT	0x00000000UL
WindowsExStyleRTLREADING	0x00002000UL
WindowsExStyleLTRREADING	0x00000000UL
WindowsExStyleLEFTSCROLLBAR	0x00004000UL
WindowsExStyleRIGHTSCROLLBAR	0x00000000UL
WindowsExStyleCONTROLPARENT	0x00010000UL
WindowsExStyleSTATICEDGE	0x00020000UL
WindowsExStyleAPPWINDOW	0x00040000UL
WindowsExStyleOVERLAPPEDWINDOW	WindowsExStyleWINDOWEDGE |WindowsExStyleCLIENTEDGE
WindowsExStylePALETTEWINDOW	WindowsExStyleWINDOWEDGE |WindowsExStyleTOOLWINDOW |WindowsExStyleTOPMOST

ICA Window Style Values

The ICA window style is an ORed combination of the following bitmasks:

ICAWindowStyle	Value
WindowIsNormal	0x00000000UL
WindowIsInActive	0x00000001UL
WindowIsMenuFlag	0x00000002UL
WindowIsIME	0x00000004UL

Methods

The functions listed below enable the programmatic access and control of the top level windows in an ICA session. Each window associated with the window object can be moved, resized, or have its Z-order modified. Use these methods with caution because some applications are intolerant of having their child windows moved or resized or their Z-order manipulated.

Method	Calling Convention	Description
BringToTop	HRESULT BringToTop( )	Requests that this window be placed at the top of the z-order and activated (that is, given input focus). If this window is not a top-level window, its parent top-level window is activated.
Resize	HRESULT Resize(Integer width, Integer height)	Requests that this window be resized to the specified width and height in pixels.
Move	HRESULT Move(Integer xPos, Integer yPos)	Requests that the window located at the top left corner of the session be moved to the region specified by coordinates.

Events

Following are a list of useful event notifications for each window associated with a window object. Use these events to track various operations performed by users on windows within an ICA session.

Method	Calling Convention	Description
OnMove	HRESULT OnMove(Integer xPos, Integer yPos)	Indicates that the window was moved to the location specified by the X and Y coordinates (xPos, yPos).
OnSize	HRESULT OnSize(Integer width, Integer height)	Indicates that a window within the ICA session was resized to the new width and height, in pixels.
OnDeactivate	HRESULT OnDeactivate ( )	The container must keep track of the focus changes using OnActivate( ) and ForegroundWindow properties. This function is not supported in Version 7.0 or earlier releases of the ICA Win32 Client. Future versions of the client will expose OnDeactivate.
OnActivate	HRESULT OnActivate ( )	Indicates that a top-level window received input focus.
OnMinimize	HRESULT OnMinimize( )	Indicates that a window was minimized.
OnCaptionChange	HRESULT OnCaptionChange(String newCaption)	Indicates that the caption on a window was changed.
OnStyleChange	HRESULT OnStyleChange(Integer style, Integer extendedStyle)	Indicates that the style bits of a window were changed.

Method	Calling Convention	Description
OnSmallIconChange	HRESULT OnSmallIconChange(String smallIconHash)	Indicates that the small icon for a window was changed. SmallIconHash is a short string representing the new icon's contents. The hash is computed from the pixel values of the icon, and is extremely unlikely to contain the same value for more than one image.
OnLargeIconChange	HRESULT OnLargeIconChange(String largeIconHash)	Indicates that the large icon for a window was changed. LargeIconHash is a short string representing the new icon's contents. The hash is computed from the pixel values of the icon, and is extremely unlikely to contain the same value for more than one image.
OnDestroy	HRESULT OnDestroy( )	Indicates that this window no longer exists within the session. The Disposed property is set to TRUE before OnDestroy is called. Hereafter all calls to methods on this object fail with E_FAIL and subsequent events will not be delivered. Property values at the time of destruction remain. Child windows receive this event and are destroyed before the parent.

Mouse

Enables the container to receive events pertaining to mouse clicks and movements. Modifier keys are specified as for keyboard.

Mouse button click states are represented by an ORed set of bitmasks, defined by the following MouseButton enumeration constants:

Constant	Value
MouseButtonLeft	1
MouseButtonRight	2
MouseButtonMiddle	4

Methods

The following methods enable programmatic insertion of mouse events into the ICA session. These methods simulate mouse events in an ICA session. If you need to emulate a typical user's mouse interaction, you can use these methods.

Method	Calling Convention	Description
SendMouseUp	SendMouseUp(Integer ButtonID, Integer modifierState, Integer xPos, Integer yPos)	Emulates a mouse button release. Note that ButtonID must specify a single mask value.
SendMouseDown	SendMouseDown(Integer ButtonID, Integer modifierState, Integer xPos, Integer yPos)	Emulates a mouse button click. Note that ButtonID must specify a single mask value.
SendMouseMove	SendMouseMove(Integer buttonState, Integer modifierState, Integer xPos, Integer yPos)	Emulates a mouse movement.

Events

The modifierState is presently unused for mouse notification messages. The values of the modifierState for these events is always 0. The following notifications are provided when the ICA Simulation subsystem is in record mode, that is, when the replayMode property of the session object is set to FALSE. Use these events to record mouse interactions between a user and the ICA session.

Event	Calling Convention	Description
OnMove	OnMove(Integer buttonState, Integer modifierState, Integer xPos, Integer yPos)	Indicates that the mouse pointer was moved to a new position (xPos, yPos) within the ICA session. The ButtonState and ModifierState are ORed combinations of bitmasks.
OnMouseUp	OnMouseUp(Integer buttonState, Integer modifierState,Integer xPos, Integer yPos)	Indicates that a mouse button was released. The ButtonState and ModifierState are ORed combinations of bitmasks. The pointer position is specified by the X and Y coordinates (xPos, yPos).
OnMouseDown	OnMouseDown(Integer buttonState, Integer modifierState, Integer xPos, Integer yPos)	Indicates that a mouse button was depressed. The ButtonState and ModifierState are ORed combinations of bitmasks. The pointer position is specified by the X and Y coordinates (xPos, yPos).
OnDoubleClick	OnDoubleClick ( )	Indicates that the mouse button was double-clicked. **Note**: Two mouse down or up events are received for each double click.

Capturing Mouse and Keyboard Events

Standard dispatch-based event handlers can be attached to the keyboard and mouse objects to receive notification of user input actions. Typically these events might be saved to data storage for later replay.

For applications that need to be aware of double-click events, you can use OnDoubleClick() in addition to the low-level OnMouseUp() and OnMouseDown() events.

Keyboard

Keyboard enables the container to receive key press or release events and notification of keyboard events received by the client.

The key IDs used correspond to the constants shared by most Microsoft scripting environments and the VK_xxxx virtual key definitions in Winuser.h. Modifier state bitmasks are defined by the following KeyModifier enumeration constants:

Constant	Value
KeyModiferShift	1
KeyModiferControl	2
KeyModiferAlt	4
KeyModifierExtended	8

If an extended key (right ALT or right CTRL) is pressed, 2 bits of the KeyModifier are set. For example, for a right ALT key, KeyModifier is to 1100 in binary.

Methods

The following functions enable programmatic insertion of keyboard events into an ICA session. These methods can be used to simulate keyboard events in an ICA session. If you need to emulate a typical user's keyboard interaction within a session, you can use these methods.

Method	Calling Convention	Description
SendKeyUp	SendKeyUp(Integer KeyID)	Emulates a key press.
SendKeyDown	SendKeyUp(Integer KeyID)	Emulates a key release.

Events

The following notifications are provided when the ICA Simulation subsystem is in Record mode; that is, when the ReplayMode property of the session object is set to FALSE. Use these events to record keyboard interactions between a user and the ICA session.

Event	Calling Convention	Description
OnKeyDown	OnKeyDown(Integer KeyID, Integer modifierState)	Indicates that a key press with the values of keyID and modifier key state was detected.
OnKeyUp	OnKeyUp(Integer KeyID, Integer modifierState)	Indicates that a key release with the values of keyID and modifier key state was detected.

ScreenShot

Specifies the region of the session screen to monitor. ScreenShot obtains either the bitmap contents or a hash representation of the bitmap contents. The specified screen region is saved in the memory mapped file each time Save( ) is called.

Properties

Property	Type	Access	Description
Filename	String	RW	Specifies the filename to store the screen snapshot to (the default value is generated automatically when Save( ) is called).
PositionX	Integer	RW	Specifies the X-coordinates (default value is the whole session).
PositionY	Integer	RW	Specifies the Y-coordinates (default value is the whole session).
Width	Integer	RW	Specifies the width of the region (default value is the whole session).
Height	Integer	RW	Specifies the height of the region (default value is the whole session).
BitmapHash	String	R	A hash computed from a bitmap contained in the screen region specified. The color depth on the desktop of the client device is set to the same as that on the server. This short hash string is generated internally using an algorithm that is highly unlikely to produce the same hash value for two different images. **Note**: This value changes whenever the region of interest is altered (the OnUpdate() event is issued). In addition, this value is dynamically generated and an error code may be returned to the user in case of error (refer to error codes applicable to Save().

Methods

Use the Save function to programmatically acquire pixel values representing the screen region associated with the ScreenShot object. Captured data is saved in Microsoft Windows BMP format to the filename specified in the object Filename property.

Method	Calling Convention	Description
Save	HRESULT Save()	Captures an image of the current session in BMP format. The BMP is stored to the file specified in the filename property. Returns one of the four error codes listed below: • **E_INVALIDARG**: If the screen size given is invalid. • **E_OUTOFMEMORY**: If the memory allocation of buffering storage fails. • **E_POINTER**: BitmapHash property operation only, if the input ScreenShot object is invalid. • **E_HANDLE**: If an invalid parameter is passed to the graphics layer. This is an internal error generated by the Citrix graphics system. • **E_FAIL**: A general error condition not classified.

Events

The OnUpdate notification is provided to indicate a change in the image state of the screen region associated with the ScreenShot object.

Event	Calling Convention	Description
OnUpdate	OnUpdate(String bitmapHash)	Indicates that the bitmap image of the specified screen region recently changed. The new hash value computed from the bitmap contents of the region is returned as BitmapHash. OnUpdate is triggered when the bitmap image of the specified screen changed within a period of one second or less.

Usage Guidelines

It is assumed that you will use the ScreenShot object to monitor a region of a session window that is in the foreground (see ICA Windows). Windows not currently in the foreground may be obscured by those higher up the Z-Order, thus overlapping the ScreenShot's region of interest (ROI).

If you use the ScreenShot object to obtain a HASH of an ROI, it is important to ensure that the client desktop resolution and color depth are identical for the record and playback of any script that uses the HASH as a sync point. It is also preferable to use a non-paletted ICA connection, that is, a 16-bit or better ICA session on a 16-bit or better client desktop.

It is also important to point out that with hash values, even a single binary bit change in a pixel value will result in a different HASH value for the ROI. In 256 color mode, all applications on the desktop share the same palette. Depending on which application on the client desktop is in the foreground, the palette might change. This changed palette might affect the calculation of BitmapHash because pixel index values in the palette may change. To alleviate this situation, it is best to have the client desktop and the ICA client session color depth in a non-paletted mode like 16bpp or 24bpp color.

ICA Client

The ICA Client Object, Version 2.3 now incorporates two additional properties as described below.

Properties

Property	Type	Access	Description
OutputMode	Enumeration {· OutputModeNonHeadless, OutputModeNormal, OutputModeRenderless, OutputModeWindowless}	RW	Specifies the ICA Simulation mode in which the client is operating. The mode could be Non-headless, Normal, Windowless, or Renderless. For example, if **ico** is an instance of the ICA Client Object, then in a C++ program, set the output mode as follows: `ico.outputMode = OutputModeWindowless`
Session	Session	R	Specifies the session object; returns NULL if the ICA session is not established.

Usage Guidelines

OutputMode must be defined only at load-time; that is, before a connection is launched.

- **OutputModeNonHeadless** = 0

  This mode disables headless functionality resulting in eliminating any overheads incurred when headless functionality is enabled. Window management functionality is also disabled. Seamless windows works only in this output mode. Selecting a mode other that OutputModeNonHeadless disables seamless windows functionality. This is the typical mode of operation when headless functionality is disabled.

- **OutputModeNormal** = 1

  This is the normal headless mode of operation. Select this mode prior to connecting to MetaFrame Presentation Server to enable the container to receive window management events and obtain a tree of windows in the session.

- **OutputModeRenderless** = 2

  The client runs as normal, does not allocate a offscreen bitmap surface, and discards all graphical data (maximum scalability). This mode results in minimal CPU and memory usage on the client device. Screen shots are not possible in this mode because the bitmap data is not available.

- **OutputModeWindowless= 3**

  The client runs as normal, but does not display in the session window. Maintains internal bitmap surface for screen snapshots. Select this mode to prevent the client from drawing to the screen if client CPU usage is identified as a bottleneck. Rendering still occurs in the background to an off-screen surface, making it possible to obtain screen captures of the session if desired.

License Agreement

This is a legal agreement (AGREEMENT) between you, the licensed user or the authorized representative of the licensed user, and Citrix Systems, Inc. (CITRIX). BY USING THE ATTACHED SOFTWARE DEVELOPMENT KIT, YOU ARE INDICATING THAT YOU ARE AUTHORIZED TO BIND THE LICENSED USER IN CONTRACT AND THAT THE LICENSED USER ACCEPTS THE TERMS OF THIS AGREEMENT. IF YOU, AS THE LICENSED USER OR REPRESENTATIVE OF THE LICENSED USER DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, DO NOT DOWNLOAD, INSTALL, COPY OR OTHERWISE USE THE CITRIX SOFTWARE DEVELOPMENT KIT.

1. **GRANT OF LICENSE**. The CITRIX software development kit includes certain example source code, tools, utilities, program interfaces and text files (collectively called the "SOFTWARE DEVELOPMENT KIT") to software that provides services on a computer called a server ("Server Software") and software that allows a computer to access or utilize the services provided by the Server Software ("Client Software"). CITRIX grants to the licensed user the following limited, non-exclusive and non-transferable rights to the SOFTWARE DEVELOPMENT KIT and accompanying documentation downloaded or otherwise obtained from Citrix:

a) **Installation and Transfer**.The licensed user may install one copy of the SOFTWARE DEVELOPMENT KIT on a single computer owned by the licensed user. The licensed user may transfer the SOFTWARE DEVELOPMENT KIT to any other computer of the licensed user, provided that it is removed from the computer from which it is transferred. The licensed user may make one (1) copy of the SOFTWARE DEVELOPMENT KIT in machine-readable form solely for backup purposes, provided that the licensed user reproduces all proprietary notices on the copy.

b) **Use of the SOFTWARE DEVELOPMENT KIT**.The licensed user may use the SOFTWARE DEVELOPMENT KIT and accompanying documentation solely to develop applications that access or utilize the Server Software.

c) **Export Notice.**The licensed user may not export or re-export the SOFTWARE DEVELOPMENT KIT in any form without the appropriate United States and foreign government licenses. The licensed user is responsible for maintaining the security of the environment in which the SOFTWARE DEVELOPMENT KIT is used. None of the SOFTWARE DEVELOPMENT KIT or underlying information or technology may be exported or re-exported (i) into (or to a national or resident of) Cuba, Iraq, Libya, Sudan, North Korea, Iran, Syria or any other country to which the United States has embargoed goods; or (ii) to anyone on the U.S. Treasury Department's list of Specially Designated Nationals or the U.S. Commerce Department's Table of Denial Orders. By installing or using the SOFTWARE DEVELOPMENT KIT, the licensed user is agreeing to the foregoing and is representing and warranting that such licensed user is not located in, under the control of, or a national or resident of any such country or on any such list. In addition, the licensed user is responsible for complying with any local laws in such licensed user's jurisdiction which may impact such licensed user's right to import, export or use the SOFTWARE DEVELOPMENT KIT.

If the SOFTWARE DEVELOPMENT KIT is identified as a North American or not-for-export product (for example, on the box, media or in the installation process), then, unless the licensed user has an exemption from the United States Department of State, the following applies: EXCEPT FOR EXPORT TO CANADA FOR USE IN CANADA BY CANADIAN CITIZENS, THE SOFTWARE DEVELOPMENT KIT AND ANY UNDERLYING TECHNOLOGY MAY NOT BE EXPORTED OUTSIDE THE UNITED STATES OR TO ANY FOREIGN ENTITY OR "FOREIGN PERSON" AS DEFINED BY U.S. GOVERNMENT REGULATIONS, INCLUDING WITHOUT LIMITATION, ANYONE WHO IS NOT A CITIZEN, NATIONAL OR LAWFUL PERMANENT RESIDENT OF THE UNITED STATES. BY DOWNLOADING OR USING THE SOFTWARE DEVELOPMENT KIT, THE LICENSED USER IS AGREEING TO THE FOREGOING AND THE LICENSED USER IS WARRANTING THAT SUCH LICENSED USER IS NOT A "FOREIGN PERSON" OR UNDER THE CONTROL OF A "FOREIGN PERSON."

d) **Other.** Notice to Users — The licensed user shall inform all users of the SOFTWARE DEVELOPMENT KIT of the terms and conditions of this AGREEMENT. Not For Resale Software - If the SOFTWARE DEVELOPMENT KIT is labeled "Not For Resale" or "NFR," this license only permits use for demonstration, test, or evaluation purposes.

2. **DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS.** The licensed user may not rent, lease, sublicense, assign or otherwise transfer or distribute the SOFTWARE DEVELOPMENT KIT. The licensed user may not modify, translate, reverse engineer, decompile, or disassemble or copy (except for the backup copy of the SOFTWARE DEVELOPMENT KIT) any part of the SOFTWARE DEVELOPMENT KIT, except to the extent such foregoing restriction is expressly prohibited by applicable law. The licensed user may not remove any proprietary notices, labels, or marks on the SOFTWARE DEVELOPMENT KIT and accompanying documentation.

THE LICENSED USER MAY NOT USE, COPY, MODIFY, OR TRANSFER THE SOFTWARE DEVELOPMENT KIT OR ANY COPY IN WHOLE OR IN PART, OR GRANT ANY RIGHTS IN THE SOFTWARE DEVELOPMENT KIT OR ACCOMPANYING DOCUMENTATION, EXCEPT AS EXPRESSLY PROVIDED IN THIS LICENSE. ALL RIGHTS NOT EXPRESSLY GRANTED ARE RESERVED BY CITRIX OR ITS SUPPLIERS.

MAINTENANCE. Citrix is not obligated to provide maintenance, updates or any other support to the licensed user with respect to the SOFTWARE DEVELOPMENT KIT and accompanying documentation licensed under this AGREEMENT.

DISCLAIMER OF WARRANTY. THE SOFTWARE DEVELOPMENT KIT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CITRIX AND ITS SUPPLIERS FURTHER DISCLAIM ALL WARRANTIES OR CONDITIONS, EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK ARISING OUT OF THE USE OR PERFORMANCE OF THE SOFTWARE DEVELOPMENT KIT AND ACCOMPANYING DOCUMENTATION REMAINS WITH RECIPIENT. CITRIX does not warrant that the SOFTWARE DEVELOPMENT KIT will be uninterrupted or error free.

SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES SO THE ABOVE EXCLUSIONS MAY NOT APPLY TO THE LICENSED USER. THIS WARRANTY GIVES THE LICENSED USER SPECIFIC LEGAL RIGHTS. THE LICENSED USER MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM STATE TO STATE.

PROPRIETARY RIGHTS. This license is not a sale. Title and copyrights to the SOFTWARE DEVELOPMENT KIT and accompanying documentation and any copy made by the licensed user remain with CITRIX or its suppliers.

LIMITATION OF LIABILITY. IN NO EVENT WILL CITRIX OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER, INCLUDING, WITHOUT LIMITATION, LOSS OF DATA, LOSS OF INCOME, LOSS OF OPPORTUNITY OR PROFITS, BUSINESS INTERRUPTION, COST OF RECOVERY OR OTHER PECUNIARY LOSS OR ANY SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF THE SOFTWARE DEVELOPMENT KIT, REFERENCE MATERIALS OR ACCOMPANYING DOCUMENTATION OF THE SOFTWARE DEVELOPMENT KIT, HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY. THIS LIMITATION WILL APPLY EVEN IF CITRIX, ITS SUPPLIERS OR AUTHORIZED DISTRIBUTORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO THE LICENSED USER.

TERMINATION. This AGREEMENT is effective until terminated. The licensed user may terminate this AGREEMENT at any time by removing the SOFTWARE DEVELOPMENT KIT from such licensed user's computer and destroying all copies of it. Unauthorized copying of the SOFTWARE DEVELOPMENT KIT or otherwise failing to comply with the terms and conditions of this AGREEMENT will result in automatic termination of this license and will make available to CITRIX other legal remedies. Upon termination of this AGREEMENT, the license granted herein will terminate and the licensed user must immediately destroy the SOFTWARE DEVELOPMENT KIT and all backup copies thereof.

GOVERNMENT END-USERS. The SOFTWARE DEVELOPMENT KIT and accompanying documentation are "commercial items," developed exclusively at private expense, consisting of "commercial computer software" and "commercial computer software documentation" as such terms are defined in the applicable acquisition regulations. If the SOFTWARE DEVELOPMENT KIT and the documentation are licensed hereunder for distribution to Government End-Users, such SOFTWARE DEVELOPMENT KIT and documentation are licensed (i) only as a commercial item, and (ii) with only those rights as are granted to all other end users pursuant to the terms and conditions of this Agreement. If this Agreement fails to meet a Government End-User's minimum needs or is inconsistent with Federal Procurement law, the licensed user agrees to notify Citrix. The following additional statement applies only to procurements governed by DFARS Subpart 227.4 (1988): "Restricted Rights æ Use, duplication and disclosure by the Government is subject to restrictions set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 (1988)."

This AGREEMENT will be governed by the laws of the State of Florida, without reference to conflict of laws principles. In any dispute arising out of this AGREEMENT, the licensed user and CITRIX each consent to the exclusive personal jurisdiction and venue in the State and Federal courts within Broward County, Florida. If either Citrix or the licensed user employs attorneys to enforce any rights arising out of or relating to this Agreement, the prevailing party shall be entitled to recover reasonable attorneys' fees and expenses.

Should you have any questions concerning this AGREEMENT, or if you desire to contact Citrix Systems, Inc. for any reason, please write: Citrix Systems, Inc., Customer Service, 851 W. Cypress Creek Road, Fort Lauderdale, Florida, 33309.