



**IntradocClient OCX
Component Reference Guide**

SDK-EN5-610

© 1996-2002 Stellent, Inc. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system without written permission from the owner, Stellent, Inc., 7777 Golden Triangle Drive, Eden Prairie, Minnesota 55344 USA. The copyrighted software that accompanies this manual is licensed to the Licensee for use only in strict accordance with the Software License Agreement, which the Licensee should read carefully before commencing use of this software.

Stellent, the Stellent logo, Stellent Content Server, Stellent Content Management, Stellent Content Publisher, Stellent Dynamic Converter, and Stellent Inbound Refinery are trademarks of Stellent, Inc. in the USA and other countries.

Adobe, Acrobat, the Acrobat Logo, Acrobat Capture, Distiller, Frame, the Frame logo, and FrameMaker are registered trademarks of Adobe Systems Incorporated.

ActiveIQ is a trademark of ActiveIQ Technologies, Incorporated. Portions Powered by Active IQ Engine.

BEA WebLogic Personalization Server is a trademark of BEA Systems, Inc.

HP-UX is a registered trademark of Hewlett-Packard Company

IBM, Informix, and WebSphere are registered trademarks of IBM Corporation.

Kofax is a registered trademark, and Ascent and Ascent Capture are trademarks of Kofax Image Products.

Linux is a registered trademark of Linus Torvalds.

Microsoft is a registered trademark, and Windows, Word, and Access are trademarks of Microsoft Corporation.

MrSID is property of LizardTech, Inc. It is protected by U.S. Patent No. 5,710,835.

Foreign Patents Pending.

Oracle is a registered trademark of Oracle Corporation.

Portions Copyright © 1991-1997 LEAD Technologies, Inc. All rights reserved.

Portions Copyright © 1990-1998 Handmade Software, Inc. All rights reserved.

Portions Copyright © 1988, 1997 Aladdin Enterprises. All rights reserved.

Portions Copyright © 1997 Soft Horizons. All rights reserved.

Portions Copyright © 1999 ComputerStream Limited. All rights reserved.

Portions Copyright © 1995-1999 LizardTech, Inc. All rights reserved.

Red Hat is a registered trademark of Red Hat, Inc.

Sun is a registered trademark, and Solaris, iPlanet, Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc.

Sybase is a trademark of Sybase, Inc.

UNIX is a registered trademark of The Open Group.

Verity is a registered trademark of Verity, Incorporated.

All other trade names are the property of their respective owners.

Table of Contents



CHAPTER 1: OVERVIEW

Introduction	1-1
About This Guide	1-2
Audience	1-2
Conventions	1-2
Stellent Product Distinctions	1-3
If You Need Assistance	1-4
Support Options	1-4
Before Contacting Support	1-5
Telephone	1-5
E-Mail	1-5
Internet	1-6

CHAPTER 2: INTRADOCCLIENT COMPONENT

Introduction	2-1
Conceptual Overview	2-2
Properties	2-2
Methods	2-2
Events	2-3
OCX Interface	2-3
OCX Integration Example	2-4
IntradocClient OCX Component Setup	2-4

CHAPTER 3: INTRADOCCLIENT API SPECIFICATIONS

Introduction	3-1
Conceptual Overview	3-2
IntradocClient OCX Properties	3-3
IntradocClient OCX Methods	3-5
AboutBox	3-5
Back	3-5
CancelRequest	3-6
DoCheckoutLatestRev	3-6
DownloadFile	3-7
DownloadNativeFile	3-8
Drag	3-9
EditDocInfoLatestRev	3-10
Forward	3-11
GoCheckinPage	3-11
Home	3-13
InitiateFileDownload	3-13
InitiatePostCommand	3-14
Move	3-15
Navigate	3-16
NavigateCgiPage	3-16
RefreshBrowser	3-17
SendCommand	3-17
SendPostCommand	3-18
SetFocus	3-19
ShowDMS	3-19
ShowDocInfoLatestRev	3-20
ShowWhatsThis	3-21
StartSearch	3-21
Stop	3-22
UndoCheckout	3-22
ViewDocInfo	3-23
ViewDocInfoLatestRev	3-24
ZOrder	3-25
IntradocClient OCX Events	3-26

APPENDIX A: UNDERSTANDING HDA FILES

Introduction	A-1
HDA File Type	A-2
HDA File Structure	A-2
Section Types	A-2
HDA Section Type: @Properties	A-3
HDA Section Type: @ResultSet	A-4
Data Binder	A-8

Chapter

1

OVERVIEW

INTRODUCTION

The information contained in this guide is based on Stellent™ Content Server 6.1. The information is subject to change as the product technology evolves and as hardware and operating systems are created and modified.

Due to the technical nature of browsers, web servers, and operating systems, Stellent, Inc. cannot warrant compatibility with all versions and features of third-party products.

This chapter contains these topics:

- ❖ About this Guide
- ❖ Stellent Product Distinctions
- ❖ If You Need Assistance

ABOUT THIS GUIDE

This document provides information on the IntradocClient OCX Component. This component is an ActiveX control used to connect to a remote Content Server and perform server functions. A description of the setup process for the component in the Microsoft Visual Basic development environment and the IntradocClient OCX API specifications listing the Properties, Methods, and Events are provided.



Audience



This guide is intended for application developers who need to access Stellent Content Server functions with an OCX interface.

Conventions

The following conventions are used throughout this guide:

- ❖ The notation *<install_dir>/* is used throughout this guide to refer to the location on your system where Stellent Content Server product is installed.
- ❖ Forward slashes (/) are used to separate the directory levels in a path name. A forward slash will always appear after the end of a directory name.
- ❖ Notes, technical tips, important notices, and cautions use these conventions:

Symbol	Description
	This is a note. It brings special attention to information.
	This is a tech tip. It identifies information that can be used to make your tasks easier.

Symbol	Description
	This is an important notice. It identifies a required step or critical information.
	This is a caution. It identifies information that might cause loss of data or serious system problems.

STELLENT PRODUCT DISTINCTIONS

In this guide, the term *content server* is used generically to refer to both the Content Server and the Collaboration Server. The following table lists the distinctions of these two Stellent content management solutions:

Stellent Content Management Product and Feature Distinction

Product	Description
Stellent Content Server	A fully functional content management system providing end-to-end content management and personalized delivery of that content.
Stellent Collaboration Server	A fully functional content management system providing end-to-end content management and personalized delivery of that content. Additionally, a Stellent Collaboration Server license enables project-level security for collaborative authoring environments.

IF YOU NEED ASSISTANCE

The Stellent family of products is backed by a full range of support options to meet every business need. The service philosophy is to keep your Stellent environment fully operational by providing the best information and solutions available. The Stellent product support team consists of highly trained product engineers who excel at resolving complex technical issues. Every customer inquiry is tracked and managed through automated systems.



Important: The support options that are available for specific systems may vary, depending on the applicable service and maintenance agreements. Please refer to your contract for the support details for your Stellent system.

Support Options

You can choose from the following three support programs offered by Stellent:

- ❖ **Standard Maintenance and Support Program:** The standard support program is available during standard business hours domestically and internationally (Monday through Friday from 8 am to 5 pm for every time zone). It provides telephone and e-mail support for troubleshooting, bug fixes, call escalation, modifications, enhancements, and updates.
- ❖ **SDK Developer Support Program:** The SDK support program is available Monday through Friday from 8 am to 5 pm (Central Time in the USA, which is -6 hours from GMT). It provides telephone and e-mail support for customers who wish to use the Software Developer's Kit (SDK) to customize their Stellent systems.
- ❖ **Extended Support Program:** The extended support program provides the standard support services 24 hours a day and 7 days a week.



Note: Value Added Resellers (VARs) and Original Equipment Manufacturers (OEMs) may have different support programs in place.

Before Contacting Support

When you call or send e-mail, please provide the following information:

- ❖ Nature and severity of the problem.
- ❖ Stellent product and version.
- ❖ Serial number of the registered Stellent product.
- ❖ Operating system and version.
- ❖ Name and telephone number of the person the support engineers should contact if they need to call back.

In addition, depending on the situation, it may be helpful to know the following:

- ❖ Database type and version.
- ❖ Web browser type and version.
- ❖ Web server type and version.

Telephone

Technical support is available from the Support Hotline at 1-888-688-TECH (1-888-688-8324). The Support Hotline is accessible toll-free world-wide.

E-Mail

The Stellent support e-mail address is *support@stellent.com*. It is available for all technical support questions.

Internet

Technical support is also available through the Internet at <http://support.stellent.com>. You will be prompted for a username and password. To obtain a username and password, contact the Support Hotline at 1-888-688-TECH (1-888-688-8324).

INTRADOCCLIENT COMPONENT

INTRODUCTION

This chapter contains these topics:

- ❖ Conceptual Overview
- ❖ OCX Interface
- ❖ OCX Integration Example

CONCEPTUAL OVERVIEW

The IntradocClient.ocx component is used to connect to a remote Stellent Content Server and perform server functions. This section provides a basic overview on Visual Basic properties, methods, and events.



Note: See Chapter 2: *IntradocClient API Specifications* for the defined properties, methods and events.

Properties

Properties describe or format an object and can be modified with code or by using the property window in the Visual Basic development environment. Properties describe the basic characteristic of an object.

For example:

- ❖ The `UserName` property provides the assigned user name.
- ❖ The `WorkingDir` property specifies the location where downloaded files are placed.

Methods

The Visual Basic Standard Controls provide methods that are common to every Visual Basic development environment. In addition, the IntradocClient OCX component provides methods that are private and unique to this specific component. These methods are used to perform or initiate an action rather than setting a characteristic.

For example:

- ❖ The `AboutBox()` method launches the About box containing product version information.

- ❖ The `GoCheckinPage` method checks in a new content item or a content item revision.

Events

Events are executed when the user or server performs an action.

For example:

- ❖ The `IntradocBrowserPost` event executes every time a user submits a form from within a browser.
- ❖ The `IntradocServerResponse` event executes after the server completes a requested action.

OCX INTERFACE

The Stellent IntradocClient OCX component is used within a Windows Visual Basic development environment to gain access to the content and content management functions within Stellent Content Server. The OCX integration is designed to call services in a visual development environment, or to connect to a remote content server.

The IntradocClient OCX component provides functionality that you can access with a method call. Methods perform actions and often return results. Information is passed to methods using parameters. Some functions do not take parameters; some functions take one parameter; some take several.

The IntradocClient OCX component requires a username and password to execute the commands. The user must have the appropriate permissions to execute the commands. Some commands will require an administrative access level, other commands may require only write permission.

Outside of the `init` and connection managing methods, all methods use the serialized HDA format for communication. The returned serialized HDA format string contains information about the success or failure of the command. The `StatusCode` will be negative if a failure occurs, and `StatusMessage` will indicate the error.



Note: Refer to the *IdcCommand Reference Guide* for additional information.

OCX INTEGRATION EXAMPLE

This integration example provides a description of the setup process for the IntradocClient OCX component.

IntradocClient OCX Component Setup

Follow these steps to set up the IntradocClient OCX component in the Microsoft Visual Basic development environment:

1. Select **Project—Components**.
2. Browse to the `IntradocClient.ocx` file on your system and click **Open**.
The `IntradocClient` module is added to the Component Controls list.
3. Ensure that the check box for *IntradocClient ActiveX Control* module is enabled and click **OK**.

An OCX control is placed in the list of controls.

4. Select the control and draw it on the Visual Basic form.
5. Add the following lines to create and initialize the control:

```
Set idcCmd=CreateObject("Idc.CommandX")
idcCmd.init("sysadmin", "c:\stellent\bin")
```


6. Use the Visual Basic development environment to create a visual interface.
See Figure 4-1 for an example.



Note: See the *IntradocClient OCX Reference Guide* for the IntradocClient OCX API specifications listing the properties, methods, and events.

The screenshot shows a Windows-style application window titled "Form1". It contains two large text areas for input and output, a text box for the bin directory, and a "Run Service" button.

Input:

```
@Properties LocalData
IdcService=ADD_USER
dName=user4
dUserAuthType=Local
@end
```

Output:

```
Content-type: text/html
Content-Length: 504

<?hda version="5.1.1 (build011203)" jcharset=Cp1252 encoding=iso-8859-1?>
@Properties LocalData
dUserAuthType=Local
IdcService=ADD_USER
alwaysSave=1
changedSubjects=userlist,1011122428327
refreshSubjects=
blFieldTypes=dUserChangeDate date,dUserArriveDate date,StatusMessage message
blDateFormat=M/d{/yy} {h:mm[:ss]} {aa}{zzz}!tAmerica/Chicago!mAM,PM
doAdminFields=1
isAdd=1
loadedUserAttributes=1
dUser=sysadmin
lsJava=1
changedMonikers=
dName=user4
copyAll=1
refreshSubMonikers=
refreshMonikers=
@end
```

Bin directory: c:\stellent\bin

Run Service

Figure 2-1 Visual Interface Example

INTRADOCCLIENT API SPECIFICATIONS

INTRODUCTION

This section provides the IntradocClient OCX API specifications and lists the properties, methods, and events. The *IntradocClient.ocx* component is used to connect to a remote content server and perform typical server functions.

This chapter contains these topics:

- ❖ Conceptual Overview
- ❖ IntradocClient OCX Properties
- ❖ IntradocClient OCX Methods
- ❖ IntradocClient OCX Events

CONCEPTUAL OVERVIEW

IntradocClient is an ActiveX control that allows a program to perform actions such as executing a service and retrieving file path information. The IntradocClient control is also a wrapper for the Microsoft Internet Explorer browser.

All methods use the serialized HDA format for communication. The returned serialized HDA format string contains information about the success or failure of the command.

The *IntradocClient.ocx* component provides functionality that can be performed with a method call. Methods perform actions and often return results.

Information is passed to methods using parameters. Some functions do not take parameters; some functions take one parameter; some take several. For example, a function with two parameters passed as strings would use this format:

```
Function(Parameter As String, Parameter As String) As  
String
```



Note: For a complete list of methods available from the core server, see *IdcCommand Reference Guide* in the SDK Documentation.

The IntradocClient OCX API enables users to write client applications to execute services. The component takes tables containing commands and parameters and calls the specified services. Execution results are passed back to the calling program.

IntradocClient OCX requires a username and password to execute the commands. The user must have the appropriate permissions to execute the commands. Some commands will require an administrative access level, other commands may require only write permission.

INTRADOCCLIENT OCX PROPERTIES

Each data item or “attribute” is implemented as a “Property” in Visual Basic. Properties are exposed through the Public Interface of an object within the Visual Basic development environment. These attributes can be used to further describe elements.

This is the IntradocClient Properties list:

Property	Description
ClientControlledContextValue	Provides the user-supplied context value. Returns the value as a string. Takes no parameters. This value becomes available to Idoc Script as the variable ClientControlled in any web page delivered by the content server.
HostCgiUrl	Provides the complete URL path of the host CGI bin. Returns the value as a string. Takes no parameters.
Password	Provides the assigned user password. Returns the value as a string. Takes no parameters.

Property	Description
UseBrowserLoginPrompt	<p>Allows the use of a browser login prompt. Defines whether a dialog box for user authentication will display.</p> <ul style="list-style-type: none"> • If set to TRUE, control will open a dialog box for user authentication • Default is TRUE. <p>Returns a Boolean value:</p> <ul style="list-style-type: none"> • Returns TRUE if the login was successful. • Returns FALSE if the login was denied.
UseProgressDialog	<p>Allows the use of a user progress dialog. Defines whether a dialog box for user authentication will display.</p> <ul style="list-style-type: none"> • If set to TRUE, control will open a dialog box for user progress. • Default is TRUE. <p>Returns a Boolean value:</p> <ul style="list-style-type: none"> • Returns TRUE if the action was completed. • Returns FALSE if the action failed.
UserName	<p>Provides the assigned user name. Returns the value as a string. Takes no parameters.</p>
WorkingDir	<p>Specifies the working directory as a full path. Returns the value as a string. Takes no parameters.</p> <p>This is the location where downloaded files are placed.</p>

INTRADOCCLIENT OCX METHODS

AboutBox

Description

Launches the About box containing product version information.

- ❖ This method performs a call and displays the product About box. The method returns FALSE if the call cannot be executed. Takes no parameters.

Definition

Public Sub

Back

Description

Displays the previous HTML page.

- ❖ Returns the user to the previous screen.
- ❖ The method retrieves the previous HTML page from cached information for display to the user. Takes no parameters.

Definition

Public Sub

CancelRequest

Description

This method cancels the currently active request. Returns **FALSE** if the function is unable to cancel the request or if there is no request currently active. Takes no parameters.

Returns a Boolean value:

- ❖ Returns **TRUE** if request is cancelled.
- ❖ Returns **FALSE** if the cancel request is not performed.

Definition

Function as Boolean

DoCheckoutLatestRev

Description

Checks out the latest content item revision.

- ❖ Given a content item name and the version label, the method checks out the latest content item revision. The function returns **FALSE** if the latest revision cannot be checked out or cannot be found in the system.
- ❖ Executes the `IntradocServerResponse` event. The event is executed before the method occurs. See *IntradocClient OCX Events* (page 3-26) for details.
- ❖ Returns serialized HDA containing `dID` and `dDocName`.
- ❖ Returns the data that was passed in as parameters.



Note: The `curID` is the content item version label not the generated content

item revision ID.

Definition

Public Sub

Parameters	Description
docName	As String The user-assigned content item name.
curID	As String The unique identifier for the latest revision.

DownloadFile

Description

Downloads the defined file.

- ❖ Given a currently-associated command and the file type, this method performs a file download of the post-conversion file (compare `DownloadNativeFile`). The function returns `FALSE` if it is unable to download the specified file.
- ❖ Executes the `IntradocBeforeDownload` event. The event is executed before the method occurs. See `IntradocClient OCX Events` (page 3-26) for details.
- ❖ Returns serialized HDA containing the status code and status method.
- ❖ Returns the data that was passed in as parameters.

Definition

Function As String

Parameters	Description
command	As String The currently-associated command.
filename	As String The file format. This is the file type such as PDF, HTM, or other supported format.

DownloadNativeFile

Description

Downloads the defined native file.

- ❖ Given a content item revision ID, a content item name, and a file type, this method performs a file download of the native file (compare `DownloadFile`). The function returns `FALSE` if it is unable to download the specified file.
- ❖ Executes the `IntradocBeforeDownload` event. The event is executed before the method occurs. See `IntradocClient OCX Events` (page 3-26) for details.
- ❖ Returns serialized HDA containing `dID` and `dDocName`.
- ❖ Returns the data that was passed in as parameters.



Note: The `id` is the generated content item revision ID, not the content item version label.

Definition

Function As String

Parameters	Description
id	As String The unique identifier for the latest revision.
docName	As String The user-assigned content item name.
filename	As String The file format. This is the file type such as DOC, RTF, or any other supported format.

Drag

Description

Begins, ends, or cancels a drag operation.

- ❖ The `Drag` method is handled the same as a Standard Control implementation.
- ❖ Refer to a Visual Basic API reference for additional information.

Definition

Public Sub

Parameters	Description
nAction	Indicates the action to perform. If you omit <code>nAction</code> , <code>nAction</code> is set to 1.

The settings for the Drag method are:

Setting	Description
0	Cancel drag operation; restore original position of control.
1	(Default) Begin dragging the control.
2	End dragging — drop the control.

EditDocInfoLatestRev

Description

Edits the content item information for the latest revision.

- ❖ ODMA related. Given a content item name, the version label, and the currently-active requested action, the method edits the content item information for the latest revision.
- ❖ The function returns FALSE if the content item information for the latest revision cannot be edited or cannot be found in the system.



Note: The curID is the content item version label not the generated content item revision ID.

Definition

Public Sub

Parameters	Description
docName	As String The user-assigned content item name.

Parameters	Description
curID	As String The unique identifier for the latest revision.
activateAction	As String Passed to ODMActivate. This can be used as Idoc Script.

Forward

Description

Displays the next HTML page.

- ❖ Moves the user to the next screen.
- ❖ This method retrieves cached information for the next HTML page for display to the user.
- ❖ Takes no parameters.

Definition

Public Sub

GoCheckinPage

Description

Checks in a new content item or a content item revision.

- ❖ Given the content item revision ID and the content item name, the function checks in a new content item or a content item revision.
- ❖ The function returns FALSE if it is unable to check in the specified file.

- ❖ This method opens the content item checkin page and enters the unique content item identifier, user-assigned content item name, and any assigned content item parameters into the associated text fields. It is also specified whether this is a new content item or a revision.
- ❖ Returns serialized HDA containing dID and dDocName.
- ❖ Returns the data that was passed in as parameters.



Note: The `id` is the generated content item revision ID not the content item version label.

Definition

Public Sub

Parameters	Description
<code>id</code>	As String The unique identifier for the latest revision.
<code>docName</code>	As String The user-assigned content item name.
<code>IsNew</code>	As Boolean Defines whether the content item to be checked in is a new content item or a revision. <ul style="list-style-type: none"> ❖ If TRUE, a new unique content item version label is assigned. ❖ Default is TRUE.
<code>params</code>	As String The parameters that pre-fill the Checkin page.

Home

Description

Returns the user to the defined home page.

- ❖ Moves the user to the home screen.
- ❖ Executes an HTML page request and displays the defined home page to the user.
- ❖ Takes no parameters.

Definition

Public Sub

InitiateFileDownload

Description

Initiates a file download.

- ❖ Given the currently-associated command and the file type, the function initiates a file download. This method initiates a file download of a specific rendition of a content item, the latest revision, or the latest released revision.
- ❖ Executes the `IntradocServerResponse` event. The event is executed before the method occurs.
- ❖ See *IntradocClient OCX Events* (page 3-26) for details.
- ❖ Returns serialized HDA containing the requested information.
- ❖ Returns the data that was passed in as parameters.

Definition

Public Function As String

Parameters	Description
command	As String The currently-associated command.
filename	As String The file format. This is the file type such as PDF, HTM, or other supported format.

InitiatePostCommand

Description

Initiates a post command.

- ❖ Initiates a service call. Given assigned post data, this method initiates a post command.
- ❖ This method automatically handles the connection to the server. If the server is available, a connection will be established. On completion of the command, the connection is closed.
- ❖ Executes the `IntradocServerResponse` event. The event is executed before the method occurs. See *IntradocClient OCX Events* (page 3-26) for details.



Note: For additional information, see *IdcCommand Reference Guide* in the SDK Documentation.

- ❖ Returns serialized HDA containing the requested information.
- ❖ Returns `StatusCode` and `StatusMessage`.
The `StatusCode` will be negative if a failure occurs, and `StatusMessage` will indicate the error.

Definition

Public Function As String

Parameters	Description
postData	As String The serialized HDA containing the command. For example, the IdcCommandX service command and any necessary service parameters.

Move

Description

Moves an object.

- ❖ The `Move` method is handled the same as a Standard Control implementation.
- ❖ Refer to a Visual Basic API reference for additional information.

Definition

Public Sub

Parameters	Description
nLeft	Specifies the horizontal coordinate for the left edge of the object. This is a single-precision value.
nTop	Specifies the vertical coordinate for the top edge of the object. This is a single-precision value.

Parameters	Description
nWidth	Specifies the new width of the object. This is a single-precision value.
nHeight	Specifies the new height of the object. This is a single-precision value.

Navigate

Description

Computes the URL path.

- ❖ Given a complete URL, this method computes the URL from the serialized HDA and returns the value as a string.
- ❖ Returns serialized HDA containing the requested information.
- ❖ Returns the data that was passed in as parameters.

Definition

Public Sub

Parameters	Description
url	As String The complete URL path.

NavigateCgiPage

Description

Computes the CGI path.

Given defined content item parameters, this method computes the CGI path from the serialized HDA and returns the value as a string.

Definition

Public Sub

Parameters	Description
params	As String The assigned content item parameters.

RefreshBrowser

Description

Refreshes the browser.

- ❖ This method refreshes the web browser and updates dynamic information.

Definition

Public Sub

SendCommand

Description

Sends instructions related to a content item.

- ❖ Given defined content item parameters, the function executes a service from the content server related to content item handling.
- ❖ Returns serialized HDA containing the requested information.

- ❖ Returns the data that was passed in as parameters.

Definition

Function As String

Parameters	Description
params	As String The CGI URL encoded parameters.

SendPostCommand

Description

Sends a post command.

- ❖ Executes a service call. Given assigned post data, this method sends a post command. This method automatically handles the connection to the server. If the server is available, a connection will be established. On completion of the command, the connection is closed.
- ❖ Executes the `IntradocBrowserPost` event. The event is executed before the method occurs. See `IntradocClient OCX Events` (page 3-26) for details.



Note: For additional information, see *IdcCommand Reference Guide* in the SDK Documentation.

- ❖ Returns serialized HDA containing the requested information.
- ❖ Returns `StatusCode` and `StatusMessage`.
The `StatusCode` will be negative if a failure occurs, and `StatusMessage` will indicate the error.

Definition

Function As String

Parameters	Description
postData	As String The serialized HDA containing the command. For example, the IdcCommandX service command and any necessary service parameters.

SetFocus

Description

Assigns the focus to a control.

- ❖ The `SetFocus` method is handled the same as a Standard Control implementation.
- ❖ Refer to a Visual Basic API reference for additional information.

Definition

Public Sub

ShowDMS

Description

Opens the HTML page associated with the Content Manager.

ODMA related. Displays the Content Manager access page in a browser.

Definition

Public Sub

ShowDocInfoLatestRev

Description

Displays the content item information for the latest revision.

- ❖ Given a content item name, the version label, and the action to perform, the function displays the content item information for the latest revision in the browser control.



Note: The `curID` is the content item version label not the generated content item revision ID.

Definition

Public Sub

Parameters	Description
<code>docName</code>	As String The user-assigned content item name.
<code>curID</code>	As String The unique identifier for the latest revision.
<code>activateAction</code>	As String The currently-active requested action.

ShowWhatsThis

Description

Displays the What's This Help topic specified for an object with the WhatsThisHelpID property.

- ❖ The ShowWhatsThis method is handled the same as a Standard Control implementation. Refer to a Visual Basic API reference for additional information.

Definition

Public Sub

Parameters	Description
Object	Specifies the object for which the What's This Help topic is displayed.

StartSearch

Description

Displays the query page in the browser control.

- ❖ Preforms browser manipulation.

Definition

Public Sub

Stop

Description

Stops the browser.

- ❖ This method stops or cancels the loading of information in the browser.

Definition

Public Sub

UndoCheckout

Description

This service reverses a content item checkout.

- ❖ Given a content item name and a version label, this service attempts to locate the content item in the system and undo the check out. The service fails if the content item does not exist in the system, if the content item is not checked out or the user does not have sufficient privilege to undo the checkout.
- ❖ Executes the `IntradocServerResponse` event. The event is executed before the method occurs.
- ❖ See IntradocClient OCX Events (page 3-26) for details.



Note: The `curID` is the content item version label not the generated content item revision ID.

Definition

Public Sub

Parameters	Description
docName	As String The user-assigned content item name.
curID	As String The unique identifier for the latest revision.

ViewDocInfo

Description

Navigates to the content item information page and displays content item information in a browser.

- ❖ Performs browser manipulation. Given a content item revision ID, the method displays content item information in a browser.



Note: The `id` is the generated content item revision ID not the content item version label.

Definition

Public Sub

Parameters	Description
id	As String The unique identifier for the latest revision.

ViewDocInfoLatestRev

Description

Navigates to the content item information page and displays content item information for the latest revision.

- ❖ Given a content item name and a version label, the method displays the content item information for the latest revision.
- ❖ Returns serialized HDA containing dID and dDocName.
- ❖ Returns the data that was passed in as parameters.



Note: The curID is the content item version label not the generated content item revision ID.

Definition

Public Sub

Parameters	Description
docName	As String The user assigned content item name.
curID	As String The unique identifier for the latest revision.

ZOrder

Description

Places a specified form or control at the front or back of the z-order within its graphical level.

- ❖ The `ZOrder` method is handled the same as a Standard Control implementation.
- ❖ Refer to a Visual Basic API reference for additional information.

Definition

Public Sub

Parameters	Description
nOrder	Specifies an integer indicating the position of the object relative to other objects. If you omit nOrder, the setting is 0.

The settings for the `ZOrder` method are:

Setting	Description
0	(Default) The object is positioned at the front of the z-order.
1	The object is positioned at the back of the z-order.

INTRADOCCLIENT OCX EVENTS

Events are executed when the user or server performs an action. These are the IntradocClient OCX events:

Event	Description
IntradocBeforeDownload	Executes before a file is downloaded. Initiates the server actions and updates required prior to a download. The event passes these parameters: <ul style="list-style-type: none">ByVal params As StringcancelDownload As Boolean
IntradocBrowserPost	Executes every time a form is submitted from within a browser. The event passes these parameters: <ul style="list-style-type: none">ByVal url As StringByVal params As StringcancelPost As Boolean
IntradocBrowserStateChange	Executes whenever the browser state changes. The event passes these parameters: <ul style="list-style-type: none">ByVal browserStateItem As StringByVal enabled As Boolean

Event	Description
IntradocRequestProgress	Executes a request for a progress report to be sent from the server. This event only occurs after a method has been called. The event passes these parameters: <ul style="list-style-type: none">ByVal <code>statusData</code> As StringByVal <code>isDone</code> As Boolean
IntradocServerResponse	Executes after the server completes a requested action. For example, after a file has been downloaded. This event handles HDA encoded data that is a response from the server. This event only occurs when an action is performed in the browser. The event passes one parameter: <ul style="list-style-type: none">ByVal <code>response</code> As String



UNDERSTANDING HDA FILES

INTRODUCTION

This chapter contains these topics:

- ❖ HDA File Type
- ❖ HDA File Structure

HDA FILE TYPE

An HDA (hyper data) file is a structured ASCII text file. This file format is designed to be compact to improve network communication. In addition, HDA files allow for persistent storage. This provides the ability to maintain consistency after the application reads in data and writes out any changes. The system creates several files specifically for this purpose. HDA files are used to define custom components that are added to Content Server. The types of resources that use HDA files are: HTML includes, environment, dynamic resource tables, and templates. There are two section types of an HDA file used during the customization process: Properties and ResultSets.

HDA FILE STRUCTURE

An HDA file contains sections that begin with *@SectionType* and end with *@end*. The two main section types in an HDA files created by the system are: *@Properties* and *@ResultSet*. When creating custom components, the *ResultSet* section type is primarily used.

Section Types

An HDA file is divided into two *tagged* sections of the form:

```
@SectionType sectionname
... Section data
@end
```

There are only two section types that are relevant to Component Architecture development: *@Properties* and *@ResultSet*. All other section tags are for internal application use only.



Note: None of the section types are mandatory and can be deleted if they are not

being used.

The purpose of the HDA file is to store data and communicate with the Content Server when a request for a Content Server service is made. Service request data is comprised of name/value pairs that are defined in the properties section of the HDA file named *LocalData*. When using applets to make a service request, the data exists in the form of a *ResultSet*.

HDA Section Type: @Properties

The *@Properties* section of an HDA file consists of a set of name/value pairs (for example, *IsJava=1*) separated by carriage return line feeds. This section type begins with *@Properties* name and ends with the syntax *@end*.

For custom component creation, the only valid name for a Properties section is *LocalData*. This is because the name/value pairs are only valid for the current HDA file. The *LocalData* section refers to data specific to this particular file.

Structure

A Properties section has the following structure:

```
@Properties LocalData
property1_name=property1_value
property2_name=property2_value
...
propertyn_name=properlyn_value
@end
```

There is no comment escape character for the Properties section of an HDA file. However, you can place comments in the file either before the start of the Properties section (*@Properties*) or after the end of the Properties section (*@end*).

An example of a Properties section is the *index.hda* file, located at *<home>/documentation/data/pages/index.hda*. This is a sample of that file:

```
@Properties LocalData
PageLastChanged=952094472723
LocationInfo=Directory,Public,
IsJava=1
refreshSubMonikers=
PageUrl=/intradoc/groups/public/pages/index.htm
LastChanged=-1
TemplatePage=DIRECTORY_PAGE
IdcService=PAGE_HANDLER
LinkSelectedIndex=0
PageName=index
HeaderText=This is a sample page. The Page Name must
remain index. The Page Properties for this index page
should be customized.
PageFunction=SavePage
dSecurityGroup=Public
restrictByGroup=1
PageType=Directory
PageTitle=Stellent Content Server Index Page
@end
```

The LocalData consists of name/value pairs. This information is only maintained during the lifetime of the request and response. Unlike information about the server environment, which rarely changes, the information for each request is dynamic. From the point of view of an HTTP request, the initial LocalData is collected from the REQUEST_METHOD, CONTENT_LENGTH, and QUERY_STRING HTTP environment variables. As the service request is processed, the values in the LocalData section will be added and changed.

HDA Section Type: @ResultSet

The *@ResultSet* section of an HDA file consists of a data representation of the results of a database query. ResultSets include serialized HDA tables.

These steps describe the page assembly process:

- ❖ Information is retrieved from the `std_page_begin`, `std_page_end`, and `std_header_sections`.
- ❖ The database is queried and the results are returned.
- ❖ The returned information is merged to complete the final page.
- ❖ A `ResultSet` becomes active during a loop of a page merge. The active `ResultSet` take precedence over any other `ResultSet`s during a value search.

The `@ResultSet` section holds a definition of a table with the number of columns on the first line, the names of the columns on the next lines and the actual row values in the same order as the columns on the last lines.

A `ResultSet` section begins with `@ResultSet` name and ends with the `@end` tag. This section enables you to define columns and rows of data (a table) when creating components. Unlike a `Properties` section, a `ResultSet` name is not limited to a single value. The `ResultSet` can be given any name. However, certain names used by the Content Server are reserved.

This table lists some of the standard `ResultSet` names that have significance to the Content Server:

ResultSet Name	Significance
Components	This file contains references to the name and location of any components you may have created.
IntradocReports	This file contains information about any reports that have been defined in the system.
IntradocTemplates	This file holds all of the default templates for the system. Do not overwrite this file.

ResultSet Name	Significance
ResourceDefinition	This file contains information about any components that you might create.
SearchResultsTemplates	This file holds information about any custom templates created for returning SearchResults to the browser.

There is no comment character for a ResultSet section of an HDA file. Blank lines must not be left between the start of a section (@ResultSet) and the corresponding end of the section (@end). Blank lines and text can only be used between sections.



Note: An HDA file is not web viewable.

Structure

A ResultSet provides the ability to define columns and rows of data. After the @ResultSet name, the number of columns that the serialized table will contain is listed. The names of each of the columns with one column name per line are then listed. Each row of the table is then defined, one column at a time, with each column value appearing on a separate line. This is an example of the file structure for a ResultSet that has n columns and m rows:

```
@ResultSet name
n
column1-name
column2-name
...
columnn-name
row1-column1-value
row1-column2-value
...
row1-columnn-value
row2-column1-value
```

```

row2-column2-value
...
row2-columnn-value
rowm-column1-value
rowm-column2-value
...
rowm-columnn-value
@end

```

Sample ResultSet

This sample depicts a ResultSet named *scores*. It contains four columns: name, game1, game2, and game3. There are four sets of data for this ResultSet:

name	game1	game2	game3
Jim	187	145	154
Joe	125	167	121
John	134	134	123
Sam	125	114	133

```

@ResultSet scores
4
name
game1
game2
game3
Jim,
187
145
154
Joe
125
167

```

```
121  
John  
134  
134  
123  
Sam  
125  
114  
133  
@end
```

Data Binder

The Content Server stores a service request internally in a Data Binder. The Data Binder manages information and organizes it into these distinct categories:

- ❖ LocalData
- ❖ ResultSets
- ❖ Environment

The Data Binder differentiates between active and non-active ResultSets during the creation of an HTML page. The Data Binder categories are used to group data to determine where the data came from and how it was created. This enables the system to determine such things as search precedence when looking up a value.

By default, when trying to evaluate the substitution of a lookup key, the data in the request is evaluated in the following order:

1. LocalData
2. Active ResultSets
3. All other ResultSets
4. Environment



Note: This precedence can be changed using Idoc Script functions.

An HDA file is a serialized Data Binder and is used for both communication and data representation. The @Properties LocalData category maps to the LocalData of the Data Binder and the @ResultSet category maps to a named result in the Data Binder.

I n d e x



A

AboutBox, 3-5
Attributes
 ClientControlledContextValue, 3-3
 HostCgiUrl, 3-3
 Password, 3-3
 UseBrowserLoginPrompt, 3-4
 UseProgressDialog, 3-4
 UserName, 3-4
 WorkingDir, 3-4

B

Back, 3-5

C

CancelRequest, 3-6
ClientControlledContextValue, 3-3
Component Setup, 2-3

D

data binder, A-8
DoCheckoutLatestRev, 3-6
DownloadFile, 3-7
DownloadNativeFile, 3-8
Drag, 3-9

E

EditDocInfoLatestRev, 3-10
E-mail
 of technical support, 1-5
Events
 IntradocBeforeDownload, 3-26
 IntradocBrowserPost, 3-26
 IntradocBrowserStateChange, 3-26
 IntradocRequestProgress, 3-27
 IntradocServerResponse, 3-27

F

Forward, 3-11

G

GoCheckinPage, 3-11

H

HDA file
 structure, A-2
 type, A-2
 data binder, A-8
 HDA section type-@ResultSet, A-4
 HDA section-type@Properties, A-3
 section types, A-2
 structure, A-2
HDA section type

Index

- @Properties, A-3
 - structure, A-3
- @ResultSet, A-4
 - sample ResultSet, A-7
 - structure, A-6
- Home, 3-13
- HostCgiUrl, 3-3

I

- InitiateFileDownload, 3-13
- InitiatePostCommand, 3-14
- Internet website of technical support, 1-6
- IntradocBeforeDownload, 3-26
- IntradocBrowserPost, 3-26
- IntradocBrowserStateChange, 3-26
- IntradocClient ActiveX Control, 2-4
- IntradocClient Component, 2-1
- IntradocClient OCX Attributes, 3-3
- IntradocClient OCX component, 2-3
- IntradocClient OCX Events, 3-26
- IntradocClient OCX Methods, 3-5
- IntradocRequestProgress, 3-27
- IntradocServerResponse, 3-27

M

- Methods
 - AboutBox, 3-5
 - Back, 3-5
 - CancelRequest, 3-6
 - DoCheckoutLatestRev, 3-6
 - DownloadFile, 3-7
 - DownloadNativeFile, 3-8
 - Drag, 3-9
 - EditDocInfoLatestRev, 3-10
 - Forward, 3-11
 - GoCheckinPage, 3-11
 - Home, 3-13
 - InitiateFileDownload, 3-13
 - InitiatePostCommand, 3-14
 - Move, 3-15
 - Navigate, 3-16

- NavigateCgiPage, 3-16
- RefreshBrowser, 3-17
- SendCommand, 3-17
- SendPostCommand, 3-18
- SetFocus, 3-19
- ShowDocInfoLatestRev, 3-20
- ShowWhatsThis, 3-21
- StartSearch, 3-21
- Stop, 3-22
- UndoCheckout, 3-22
- ViewDocInfo, 3-23
- ViewDocInfoLatestRev, 3-24
- Zorder, 3-25
- Microsoft Visual Basic, 2-4
- Move, 3-15

N

- Navigate, 3-16
- NavigateCgiPage, 3-16

O

- OCX Interface, 2-3
- Overview
 - Audience, 1-2
 - Conventions, 1-2
 - Stellent Product Distinctions, 1-3

P

- Password, 3-3

R

- RefreshBrowser, 3-17
- ResultSet, sample, A-7

S

- sample ResultSet, A-7

section types (*HDA file structure*), A-2

SendCommand, 3-17

SendPostCommand, 3-18

SetFocus, 3-19

ShowDocInfoLatestRev, 3-20

ShowWhatsThis, 3-21

StartSearch, 3-21

Stop, 3-22

Support

- e-mail address, 1-5

- Internet website, 1-6

- telephone number, 1-5

- website, 1-6

Support Hotline, 1-5

T

Technical support

- e-mail address, 1-5

- telephone number, 1-5

- website, 1-6

Telephone number of technical support, 1-5

U

Understanding HDA Files, A-1

- HDA file type, A-2

UndoCheckout, 3-22

UseBrowserLoginPrompt, 3-4

UseProgressDialog, 3-4

UserName, 3-4

V

ViewDocInfo, 3-23

ViewDocInfoLatestRev, 3-24

Visual Basic, 2-4

W

Website for technical support, 1-6

WorkingDir, 3-4

Z

ZOrder, 3-25

