



**IdcCommand—Java Command
Utility Reference Guide**

SDK-EN1-610

© 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: IDCCommand: JAVA COMMAND UTILITY

Introduction	2-1
Conceptual Overview	2-2
Command Line Options	2-3
Configuration File Options	2-5
IdcCommandFile	2-6
IdcCommandUserName	2-6

Table of Contents

- IdcCommandLog2-6
- ConnectionMode (optional)2-6
- Command File Syntax2-7
- Precedence2-7
- Special Characters2-7
- Special Tags2-8
- Creating and Executing IdcCommand Parameters2-9
- Run IdcCommand on NT2-11
- Run IdcCommand on Solaris2-11
- Using the Stellent Launcher2-13
- User Interface2-13
- Configuring the Launcher2-14
- Configuration File Example2-14
- Calling Services Remotely2-18

**CHAPTER 3: IDC_COMMANDX:
ACTIVE_X COMMAND UTILITY**

- Introduction3-1
- Using IdcCommandX3-2
- IdcCommandX Setup3-2
- Calling IcdCommandX3-2
- Visual Basic3-2
- Visual C++3-2
- Executing Services3-3
- Methods3-4
- init3-4
- executeCommand3-5
- connectToServer3-6
- closeServerConnection3-7
- computeWebFilePath3-8
- computeNativeFilePath3-9
- computeURL3-10

CHAPTER 4: USING SERVICES

Introduction	4-1
Conceptual Overview	4-1
Service Request Process	4-2
Search Service Request Process	4-3
Service Structure	4-3
Service Name	4-4
Service Attributes	4-5
Service Type	4-6
Security or Access Level	4-7
Template page	4-8
Sub-Service	4-8
Subjects Notified	4-9
Service Error Message	4-9
Associated Actions	4-9
Example Service: DOC_INFO	4-12
Service Definition File	4-12
Service Template File	4-13
Service Actions	4-20
Tutorial: Creating a Custom Service	4-22
Overview	4-22
Steps	4-22
Tutorial: Creating a Query and Service Component	4-25
Overview	4-25
Steps	4-25
Custom Application Example	4-30
Sample Code	4-31

CHAPTER 5: IDCCOMMAND SERVICES

Introduction	5-1
Conceptual Overview	5-2
Admin Server Related Services	5-2
Content Publisher Related Services	5-3

Table of Contents

Customization Related Services	5-3
Personalization Related Services	5-5
Problem Report Related Services	5-6
Provider Related Services	5-7
Available Services	5-8
ADD_ALIAS	5-8
ADD_ARCHIVE	5-10
ADD_COLLECTION	5-12
ADD_DOC_ACCOUNT	5-14
ADD_DOCEXTENSION	5-15
ADD_DOCFORMAT	5-16
ADD_DOCTYPE	5-18
ADD_EDIT_PROVIDER	5-19
ADD_GROUP	5-21
ADD_METADEF	5-23
ADD_PROBLEMREPORT	5-24
ADD_ROLE	5-26
ADD_USER	5-27
ADD_WF_TEMPLATE	5-32
ADD_WORKFLOW	5-33
ADD_WORKFLOWALIAS	5-35
ADD_WORKFLOWDOCUMENT	5-37
ADD_WORKFLOWDOCUMENTS	5-39
ADD_WORKFLOWSTEP	5-41
APPLET_DOCINFO	5-45
CANCEL_ARCHIVE	5-46
CANCEL_SEARCH_INDEX	5-47
CHECK_USER_CREDENTIALS	5-48
CHECKIN_ARCHIVE	5-50
CHECKIN_ARCHIVE_NO_NOTIFY	5-57
CHECKIN_BYNAME	5-58
CHECKIN_LIST	5-65
CHECKIN_NEW	5-66
CHECKIN_NEW_FORM	5-73
CHECKIN_SEL	5-74
CHECKIN_SEL_FORM	5-81
CHECKIN_UNIVERSAL	5-82

CHECKOUT	5-89
CHECKOUT_BY_NAME.....	5-91
CHECKOUT_OK.....	5-93
CONFIG_INFO	5-94
CONTROL_SEARCH_INDEX.....	5-95
COPY_ARCHIVE	5-104
CREATE_SUBSCRIPTION_TYPE.....	5-105
CRITERIAWORKFLOW_DISABLE.....	5-107
CRITERIAWORKFLOW_ENABLE	5-108
DEL_METADEF	5-109
DELETE_ALIAS	5-110
DELETE_ARCHIVE	5-111
DELETE_BATCH_FILE	5-113
DELETE_BATCH_FILE_DOCUMENTS.....	5-115
DELETE_DOC	5-117
DELETE_DOC_ACCOUNT	5-118
DELETE_DOCEXTENSION	5-119
DELETE_DOCFORMAT.....	5-120
DELETE_DOCTYPE.....	5-121
DELETE_GROUP.....	5-122
DELETE_PROBLEMREPORT	5-123
DELETE_PROVIDER.....	5-124
DELETE_RESULT_TEMPLATE.....	5-125
DELETE_REV.....	5-126
DELETE_REV_EX	5-127
DELETE_ROLE	5-128
DELETE_SUBSCRIPTION_TYPE	5-129
DELETE_USER	5-130
DELETE_WF_TEMPLATE	5-131
DELETE_WFCONTRIBUTORS	5-132
DELETE_WORKFLOW.....	5-134
DELETE_WORKFLOWCRITERIA	5-135
DELETE_WORKFLOWDOCUMENTS	5-136
DELETE_WORKFLOWSTEP	5-137
DOC_FORMATS_WIZARD	5-139
DOC_INFO	5-140
DOC_INFO_BY_NAME	5-141

Table of Contents

DOC_INFO_LATESTRELEASE	5-142
DOC_SUBS_LIST.....	5-143
DOWNLOAD_COMPONENT	5-144
EDIT_ALIAS	5-145
EDIT_ARCHIVE	5-147
EDIT_ARCHIVEDATA	5-148
EDIT_DOC_FORMATS	5-149
EDIT_DOCEXTENSION.....	5-151
EDIT_DOCFORMAT.....	5-152
EDIT_DOCTYPE.....	5-154
EDIT_EXPORTERS	5-155
EDIT_GROUP.....	5-157
EDIT_METADEF.....	5-158
EDIT_ROLE	5-161
EDIT_USER	5-162
EDIT_USER_PROFILE.....	5-167
EDIT_WF_TEMPLATE.....	5-169
EDIT_WORKFLOW	5-172
EDIT_WORKFLOWCRITERIA	5-174
EDIT_WORKFLOWSTEP.....	5-176
ENABLE_DISABLE_PROVIDER	5-180
EXPORT_ARCHIVE.....	5-182
FORM_PROCESS	5-185
FORM_SUBMIT	5-186
GET_ACTIVE_WORKFLOWS	5-187
GET_ADD_EDIT_PROVIDER_FORM.....	5-188
GET_ADMIN_PAGE.....	5-191
GET_ALIASES	5-192
GET_ALL_PROVIDERS.....	5-193
GET_ARCHIVECOLLECTIONS	5-194
GET_ARCHIVED_FILE.....	5-195
GET_ARCHIVES	5-198
GET_BATCH_FILE_DOCUMENTS	5-199
GET_BATCH_SCHEMA.....	5-201
GET_BATCH_VALUES	5-203
GET_BATCHFILES.....	5-205
GET_CRITERIA_WORKFLOWS_FOR_GROUPS.....	5-207

Table of Contents

GET_DATARESULTSET	5-209
GET_DOC_CONFIG_INFO	5-210
GET_DOC_PAGE.....	5-212
GET_DOC_SUBSCRIBERS.....	5-214
GET_DOCEXTENSIONS	5-215
GET_DOCFORMATS.....	5-216
GET_DOCTYPES.....	5-217
GET_DOCUMENT_PROBLEMREPORTS.....	5-218
GET_DYNAMIC_CONVERSION	5-219
GET_DYNAMIC_PAGE	5-223
GET_DYNAMIC_URL.....	5-224
GET_EXTERNAL_DOC_INFO.....	5-225
GET_EXTERNAL_HIGHLIGHT_INFO	5-227
GET_EXTERNAL_XML_HIGHLIGHT_INFO	5-229
GET_FILE.....	5-231
GET_FILE_BY_NAME	5-235
GET_FILELIST	5-239
GET_HIGHLIGHT_INFO.....	5-240
GET_METADEFS.....	5-243
GET_OPTION_LIST.....	5-244
GET_PORTAL_PAGE	5-245
GET_PROBLEMREPORTS_SEARCH_FORM	5-247
GET_PROBLEMREPORTS_SEARCH_RESULTS	5-248
GET_PROJECT_INFO.....	5-250
GET_PROJECT_REGISTRATION.....	5-251
GET_PROVIDER_INFO.....	5-252
GET_PROXIED_ARCHIVECOLLECTIONS.....	5-254
GET_REGISTEREDPROJECTS	5-255
GET_REPLICATION_DATA.....	5-256
GET_RESULT_OPTIONS	5-257
GET_SEARCH_RESULTS.....	5-258
GET_SECURE_PAGE	5-260
GET_SELF_REGISTER_PAGE.....	5-261
GET_SERVER_OUTPUT.....	5-262
GET_TABLE	5-264
GET_UPDATE_FORM.....	5-265
GET_UPDATE_PROBLEMREPORT_FORM.....	5-266

Table of Contents

GET_USER_INFO	5-267
GET_USERS	5-268
GET_WF_TEMPLATE	5-269
GET_WF_TEMPLATES	5-270
GET_WORKFLOW	5-271
GET_WORKFLOW_INFO	5-272
GET_WORKFLOWDOCREVISIONS	5-274
GET_WORKFLOWDOCUMENTS	5-275
GET_WORKFLOWS	5-276
GET_XML_HIGHLIGHT_INFO	5-277
IMPORT_ARCHIVE	5-279
IMPORT_DOCUMENT	5-282
LOAD_GLOBALINCLUDES	5-284
LOAD_RESOURCE_FILE	5-285
LOGIN	5-288
MERGE_TABLE	5-290
NOTIFY_CHANGE	5-294
NOTIFY_CONTRIBUTOR	5-295
PAGE_HANDLER	5-296
PING_SERVER	5-298
PNE_GET_SEARCH_RESULTS	5-300
PNE_SAVE_QUERY	5-301
PNE_UPDATE_PERSONAL_URLS	5-302
PNE_UPDATE_PORTAL_INFO	5-304
PROBLEMREPORT_INFO	5-305
QUERY_DOC_ACCOUNTS	5-306
QUERY_GROUP	5-307
QUERY_USER_ATTRIBUTES	5-308
REGISTER_IMPORTER	5-309
REGISTER_USER	5-311
REMOVE_COLLECTION	5-313
REMOVE_EXPORTER	5-315
REMOVE_IMPORTER	5-316
REQUEST_SECURITYINFO	5-317
RESEND_PROBLEMREPORT	5-318
RESUBMIT_FOR_CONVERSION	5-319
REV_HISTORY	5-320

Table of Contents

SAVE_GLOBALINCLUDES	5-321
SELECTDOC	5-322
START_SEARCH_INDEX	5-323
SUBMIT_HTML_FORM	5-325
SUBSCRIBE	5-327
SUBSCRIBE_DOC_USER	5-329
SUBSCRIBE_EX	5-331
SUBSCRIPTION_LIST	5-333
TEST_PROVIDER	5-334
UNDO_CHECKOUT	5-335
UNDO_CHECKOUT_BY_NAME	5-337
UNSUBSCRIBE	5-339
UNSUBSCRIBE_FROM_LIST	5-340
UNSUBSCRIBE_FROM_LIST_EX	5-341
UPDATE_DOCINFO	5-343
UPDATE_DOCINFO_BYFORM	5-346
UPDATE_DOCINFO_WITH_FILE	5-349
UPDATE_META_TABLE	5-352
UPDATE_OPTION_LIST	5-354
UPDATE_PROBLEMREPORT	5-356
UPDATE_RESULT_TEMPLATE	5-358
UPDATE_SUBSCRIPTION_NOTIFY	5-360
UPDATE_SUBSCRIPTION_TYPE	5-361
UPDATE_SUBSCRIPTION_USED	5-363
UPDATE_USEROPTION_LIST	5-365
UPLOAD_NEW_COMPONENT	5-367
VIEW_DOC	5-368
WORK_IN_PROGRESS	5-370
WORKFLOW_APPROVE	5-371
WORKFLOW_CANCEL	5-372
WORKFLOW_CHECKIN	5-373
WORKFLOW_REJECT	5-376
WORKFLOW_REJECT_FORM	5-377
WORKFLOW_START	5-378

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 guide on using the IdcCommand and IdcCommandX utilities to access to the content and content management functions within Stellent Content Server. The Stellent IdcCommand Java Command Utility is a stand-alone Java application that enables users to execute content server services.

Audience

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

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).

Chapter

2

IdcCOMMAND: JAVA COMMAND UTILITY

INTRODUCTION

This chapter contains these topics:

- ❖ Conceptual Overview
- ❖ Command Line Options
- ❖ Configuration File Options
- ❖ Command File Syntax
- ❖ Creating and Executing IdcCommand Parameters
- ❖ Calling Services Remotely

CONCEPTUAL OVERVIEW

The IdcCommand utility is a stand-alone Java application that enables users to execute content server services. The program reads a command file containing commands and parameters and calls the specified services. A log file records the time that the call was executed, whether the command was successfully executed, and if there were execution errors.

The IdcCommand utility requires a command file and content server username to execute the commands. The user must have appropriate permissions to execute the commands. Some commands will require administrative access, other commands may require only write permission.

Command options can be specified on the command line or in the *intradoc.cfg* configuration file. Specifying a command line option overrides the setting in the configuration file. See “Command Line Options” on 2-3 and “Configuration File Options” on 2-5 for additional information.

- ❖ The `IdcCommandFile` specifies the file containing the commands to execute. The command line option `-f` overrides this parameter.
- ❖ The `IdcCommandUserName` specifies the user permitted to run the command. The command line option `-u` overrides this parameter.
- ❖ The `IdcCommandLog` specifies the path to the log file. The command line option `-l` overrides this parameter.
- ❖ The `ConnectionMode` specifies the connection mode for executing the commands. The command line option `-c` overrides this parameter.
 - If this option is not specified, the default connection mode of `auto` is used. In the default connection mode, IdcCommand will initially attempt to connect to the server.
 - If this fails, the commands are executed in stand-alone mode.

There are certain commands that cannot be executed in stand-alone mode. In general, these commands are performed asynchronously by the server in a background thread. This happens in the update or rebuild of the search index.

Command File Syntax uses precedence to resolve conflicts among the name/value pairs within the `LocalData` section. Special characters specific to the command file syntax are used.

The `IdcCommand` utility only returns information about the success and failure of the command. To retrieve information from the server in an interactive session, use the Java COM wrapper `IdcCommandX` available on NT.



Note: See “Chapter 2: `IdcCommandX`” for additional information.

COMMAND LINE OPTIONS

Options can be specified on the command line or in the *intradoc.cfg* configuration file. Specifying a command line option overrides the setting in the configuration file.

For example:

```
-f newfile.hda -u admin -l C:/stellent/newlog.txt -c server
```



Note: See “Configuration File Options” on 2-5 for additional information.

-f command_file

Specifies the file containing the commands to execute. Use this parameter if the command file has the same name. Alternatively, this can be specified in the *intradoc.cfg* configuration file by setting the value of `IdcCommandFile`. Specifying the log file path on the command line overrides the setting in the *intradoc.cfg* file.

-u user_name

Specifies the user permitted to run the command. Alternatively, this can be specified in the *intradoc.cfg* configuration file by setting the value of `IdcCommandUserName`. Specifying the log file path on the command line overrides the setting in the *intradoc.cfg* file.

-l log_file

Specifies the path to the log file. As each command is executed, a message is sent to the log file. The log file contains the time the command was executed and the success or failure status. The log file can be used to relay processing information to the user. Alternatively, this can be specified in the *intradoc.cfg* configuration file by setting the value of `IdcCommandLog`. Specifying the log file path on the command line overrides the setting in the *intradoc.cfg* file.

- ❖ If the action performed is successful, a “success” message is written to the log file.
- ❖ If the action performed is not successful, an error message is written to the log file.
- ❖ If no log file is specified, information is logged to the screen.

-c connection_mode

Specifies the connection mode for executing the commands. If this option is not specified, the default connection mode of `auto` is used. Alternatively, this can be specified in the *intradoc.cfg* configuration file by setting the value of

`ConnectionMode`. Specifying the connection mode on the command line overrides the setting in the *intradoc.cfg* file.

Connection Mode	Description
auto	This is the default connection mode. This specifies that IdcCommand will initially attempt to connect to the server. If this fails, the commands are executed in stand-alone mode.
server	This specifies that IdcCommand will only execute services through the server.
standalone	This specifies that IdcCommand will execute the services in a stand-alone session.

CONFIGURATION FILE OPTIONS

Options can be specified on the command line or in the *intradoc.cfg* configuration file. Specifying a command line option overrides the setting in the configuration file. See “Command Line Options” page 2-3 for additional information.

For example:

```
IdcCommandFile=newfile.hda
IdcCommandUserName=sysadmin
IdcCommandLog=C:/stellent/newlog.txt
ConnectionMode=server
```



Note: See “Command Line Options” page 2-3 for additional information

IdcCommandFile

Specifies the file containing the commands to execute. Use this parameter if the command file will always have the same name. The command line option `-f` overrides this parameter.

IdcCommandUserName

Specifies the user permitted to run the command. The command line option `-u` overrides this parameter.

IdcCommandLog

Specifies the path to the log file. The log file contains the time the command was executed, and the success or failure status. If no log file is specified, information is logged to the screen. The log file can be used to relay processing information to the user. As each command is executed, a message will be sent to the log file. If the action performed is successful, a “success” message is displayed. If the action performed is not successful, an error message and related information is written to the log file. The command line option `-l` overrides this parameter.

ConnectionMode (optional)

Specifies the connection mode for executing the commands. If this option is not specified, the default connection mode of `auto` is used. The command line option `-c` overrides this parameter.

Connection Mode	Description
auto	This is the default connection mode. This specifies that IdcCommand will initially attempt to connect to the server. If this fails, the commands are executed in stand-alone mode.

Connection Mode	Description
server	This specifies that IdcCommand will execute services only through the server.
standalone	This specifies that IdcCommand will execute the services in a stand-alone session.

COMMAND FILE SYNTAX

Precedence

The command file syntax recognizes the *hda* syntax and uses precedence to resolve conflicts among the name/value pairs within the LocalData section. When normal name/value pairs are parsed, they are assumed to be within the @Properties LocalData tag. If the section contains *hda* tags, the normal name/value pairs take precedence over name/value pairs within the @Properties LocalData tag.

For example, if `foo=x` is in a normal name/value pair and `foo=y` is within the @Properties LocalData tag, the name/value pair `foo=x` takes precedence because it is outside the tag.

Special Characters

These special tags are specific to the command file syntax:

Special Characters	Description
<<EOD>>	The end of data marker. The command file is made up of one or more sections separated with an end of data marker.

Special Characters	Description
=	Within each section there are one or more name/value pairs. Each name/value pair resides on a separate line and the name is separated from the value with an equal character.
#	The comment line marker. The pound character placed at the beginning of a line indicates that the line is a comment.
\	The backslash is an escape character.



Note: The command file recognizes *hda* syntax. Each section of the command file can contain an *hda* file. Consequently, the command file is able to understand the @ResultSet and @Properties LocalData tags.

Special Tags

These special tags are specific to the command file syntax:

Special Tags	Description
IdcService=	Each section of the command file must specify the name of the service it is calling. The IdcService tag is a name/value pair that specifies the command that is to be executed. The format <code>IdcService=service_name</code> is used.

Special Tags	Description
@Include	This tag allows the user to include content from another file at the spot where the @Include tag is placed. This tag can be used to include a complete <i>hda</i> file or to include shared name/value pairs. This inclusion takes the exact content of the specified file and places it in the location of the @Include tag. A file can be included as many times as desired and an included file may include other files. However, circular inclusions are not allowed. The format @Include filename (with space) is used.



Note: Data from one section is not carried over to the next section. Each section must contain a complete set of data for the command.

CREATING AND EXECUTING IDCCommand PARAMETERS

The IdcCommand utility is run from the command line. In general, the user name and the command file must be specified. These options can be specified either on the command line or in an *intradoc.cfg* configuration file.

See the “Command Line Options” on 2-3 and “Configuration File Options” on 2-5 for an explanation of available commands.

Perform the following steps to run IdcCommand from the command line.

1. Create a new directory.

This directory will be used as the IdcCommand working directory. Use this directory for your configuration file and any command files.

2. Copy the *intradoc.cfg* configuration file from *stellent/bin/* into the newly created directory.



Important: Do not delete the WebBrowserPath or IntradocDir information.

In this example, two comment lines, the complete path to the assigned web browser, the complete path to the assigned HTML editor, and the complete path to the content server directory are defined.

```
#Configuration
#Fri Dec 31 23:59:59 CST 1999
WebBrowserPath=C:/Program Files/Plus!/Microsoft
Internet/Iexplore.exe
HTMLEditorPath=C:/Program Files/Windows
NT/Accessories/Wordpad.exe
IntradocDir=C:/stellent
```

3. Define the parameters listed in “Configuration File Options” on 2-5 in the copied *intradoc.cfg* configuration file located in the newly created directory, define the configuration file option parameters. See “Configuration File Options” on 2-5 for additional information.

In this example, IdcCommandFile references *newfile.hda*. The IdcService must be defined in that file.

```
IdcCommandFile=newfile.hda
IdcCommandUserName=sysadmin
IdcCommandLog=C:/stellent/newlog.txt
```

4. Create a command file in the newly created directory and add the desired command.

In this example, a command file named *newfile.hda* is created and a user called “Jennifer” is defined within that file. Additional samples can be found in the descriptions of the various commands.

```
# Add a user
IdcService=ADD_USER
dName=Jennifer
dUserAuthType=Local
dFullName=Jennifer Smith
```

```
dPassword=password  
dEmail=email@email.com
```

5. Run IdcCommand. Follow the steps to run IdcCommand on NT or to run IdcCommand on Solaris provided on the following pages.

Run IdcCommand on NT

To run IdcCommand on NT, follow these steps:

6. From the command line, change to the IdcCommand working directory (created in step 1).

For example:

```
cd <working directory>
```

7. From the command line, change the *jview* command to define the correct path of the virtual machine.

For example:

```
jview /cp:a <install_dir>/shared/classes/server.zip  
IdcCommand
```

Run IdcCommand on Solaris

To run IdcCommand on Solaris, follow these steps:

6. From the command line, change to the IdcCommand working directory (created in step 1).

For example:

```
cd <working directory>
```

7. From the command line, change the */bin/java* command to define the correct path of the virtual machine.

For example:

```
bin/java -classpath ./classes:/classes/idcserver3.zip:<Oracle-JDBC-Driver-Path> IdcCommand -f command-file-name
```



Note: These commands are similar to the ones defined in the batch files located in the *stellent/bin* directory.



Tech Tip: The standard command line to run an IdcCommand batch is lengthy. To simplify the process, included in the product executable are *wrappers* that can be used to execute IdcCommand with less command line entries. Compare the following two command lines:

Without *wrapper*:

```
C:\stellent\shared\os\win32\jdk1.3.1_01\bin\java -cp
%stellent%\shared\classes\server.zip idccommand -u
sysadmin -f addusers.txt -l idclog.txt
```

With *wrapper*:

```
idccommand -u sysadmin -f addusers.txt -l idclog.txt
```

The command *wrappers* are available for both Unix and Windows. They make IdcCommand more accessible and easy to execute. See *Using the Stellent Launcher* (page 2-13) for additional information.

USING THE STELLENT LAUNCHER

The Launcher is a native C++ application that allows a Java program to start as an NT service. This allows Stellent applications and servers to run from different Java classes, even if they are on the same machine. If you want to install a custom application and launch the program, you can use the Launcher.exe file to launch that program as a service.

The Launcher must reside in a directory with a valid intradoc.cfg configuration file and must have the same name as the Java class file to be launched (case sensitive).

For example:

```
<install_dir>/bin/
```



Tech Tip: By using the Launcher.exe, changing the status.dat file, and altering the value of the JVM command line, you could theoretically run any Java program as an NT service. This is not recommended for normal use, but it does explain the many ways you could configure the Launcher.

User Interface

The UI for the Launcher is exactly the same as the application it launches. For example, if the Launcher is renamed to IntradocApp, then to launch the Web Layout Editor, the following command line arguments are given:

```
IntradocApp WebLayout
```

This will launch the Web Layout Editor as a stand-alone application.

By default, the application will be launched without console output. However, when launching IdcServer, IdcAdmin, IdcCommandX, or the Installer, Java output is printed to the screen. In all other cases, the output is suppressed for a cleaner interface.

For some applications, such as the Batch Loader and the Repository Manager, it is desirable to view the Java output from the application. To force the Launcher to dump the Java output to the screen, use the “-console” flag in this manner:

```
IntradocApp RepMan -console
```

The output will now be written to the console from which the Repository Manager was launched.

If the Launcher is renamed IdcServer, BatchLoader, SystemProperties, or any other Java class that requires no additional parameters, it can be launched with a simple double-click. In other cases, a shortcut can be used to launch them by double-clicking.

Configuring the Launcher

To use the Launcher, you must first rename the Launcher.exe file to an executable with the same name as the class file to be launched. Typical examples include IdcServer.exe or IntradocApp.exe.



Note: If you want to make a custom application, you should create the custom directory, and re-name the Launcher.exe to the service that is to be launched. A valid intradoc.cfg file needs to be in the same directory as the executable. The only required parameter is IntradocDir; however, other entries can be included to alter the way the Java application is launched.

Configuration File Example

Configuration file example entries:

```
#intradoc.cfg values:  
IntradocDir=c:/stellent/  
#default additional values given below  
CLASSPATH=$COMPUTEDCLASSPATH
```

```
JvmCommandLine=c:/winnt/jview.exe /cp:a "$CLASSPATH"
$STARTUPCLASS

JvmClasspathSeparator=;
```

If the values CLASSPATH, JvmCommandLine, or JvmClasspathSeparator are not specified in the file, the default values are used. The value for COMPUTEDCLASSPATH is:

```
/classes
$SHAREDDIR/classes
$SHAREDDIR/classes/server.zip
$SHAREDDIR/classes/idcrefinery.zip
$SHAREDDIR/os/win32/classes/dao.zip
$SHAREDDIR/classes/jspserver.jar
$SHAREDDIR/j2sdk/lib/tools.jar
```

This is sufficient to launch nearly all Stellent applications. Others, such as the Inbound Refinery, require additional classes in the classpath. This file can also be modified to enable Intradoc to be run with different Java virtual machines.



Note: The intradoc.cfg file is usually altered to include JDBC drivers for their particular database upon install. If you want to use alternate JDBC drivers, place them in the <install_dir>/shared/classes directory, and alter the intradoc.cfg file accordingly.

For example, to run the Intradoc Server with IBM's virtual machine on NT, the command line would look like:

```
#customized for running IBM's VM
JvmCommandLine=jre -cp $CLASSPATH $STARTUPCLASS
```

The classpath is designed to look for class files in order of the listed entries. In other words, the Launcher will search the entire <install_dir>/classes/ directory before it looks into the shared directory or server.zip file. This is desirable if the

users wish to overload Java classes without patching the zip file. Additionally, the

Launcher can be used to install, uninstall, and run Java applications as NT Services, provided that they follow the correct API for communicating back to the Launcher. See the source code for IdcServer.java or IdcAdmin.java for more details on how to make any Java application run as an NT service with the Launcher.

These are the common execution flags run on the command line:

Console Flags	Description
-console	Forces the Launcher to print the Java output and error streams to the console.
-install	Used to install the Java application referred to by the Launcher as an NT Service.
-uninstall	Used to uninstall the Java application referred to by the Launcher as an NT Service.
-remove	Same as -uninstall.
-dependent <xxx>	<p>Makes the NT service dependent on whether or not the service <xxx> is also running.</p> <p> Note: This command is useful when you want to make a dependent call for each service.</p> <p>If you want to launch a database before starting the Content Server, you can specify the database start-up before the Content Server.</p>

Console Flags	Description
<p>-dependent <yyy> <zzz></p>	<p>Must be used along with -install. It will install the service the credentials of user <yyy> with password <zzz>.</p> <p> Note: This command will check the user regardless of the credentials, but may not install the service. The credentials of the user need to extend to the service for the auto-start to run the service automatically.</p> <p>For certain services, such as the Inbound Refinery, the last flag is required so the service can run with higher permissions. The user name must be in the typical Microsoft format DOMAIN\User. Once users change passwords, the service will not be able to log in, and therefore will not run.</p>



Tech Tip: To customize the class path to alter the system path to load up Stellent .dll files, you can change the pathway to:

```
PATH=$SHAREDDIR\os\win32\lib\;$SHAREDDIR\search\vd
k\_nti40\bin;$SHAREDDIR\search\vdk\_nti40\filters;
$OLDPATH
```

If you want to load up Stellent-specific custom .dlls, you should put them in the directory:

```
<install_dir>/shared/os/win32/lib
```

CALLING SERVICES REMOTELY

To use services remotely, you must have these two files on the remote machine:

- ❖ <instal_dir>\bin\intradoc.cfg (same file as on the content server).
- ❖ <instal_dir>\config\config.cfg

In addition, these configuration entries must be defined in the #Additional Variables section of the config.cfg file on the remote machine:

- ❖ IntradocServerPort=4444
- ❖ IntradocServerHostName=IP or DNS

IdcCOMMANDX: ACTIVE X COMMAND UTILITY

INTRODUCTION

IdcCommandX is an ActiveX control that allows a program to execute a service and retrieve file path information. IdcCommandX serves as a COM wrapper for the standard IdcCommand services used by content server.



Note: See “Chapter 1: IdcCommand” for additional information.

This chapter contains these topics:

- ❖ Using IdcCommandX
- ❖ Methods

USING IDCCommandX

This section provides the steps required the use IdcCommandX and teh procedure to call services.

IdcCommandX Setup



Note: A Visual Basic or Visual C++ development environment is required to use IdcCommandX.

Run the IdcCommandX ActiveX Command Utility setup executable located in the Extras/IDCCommandX sub-directory on the Stellent Content Server CD-ROM.

Calling IcdCommandX

Visual Basic

Add IdcCommandX as a control to the Visual Basic project and code the following lines to create and initialize the control:

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

Visual C++

Add the IdcCommandX control to the project and call the desired IdcCommandX class.

Executing Services

When executing services using the Stellent IdcCommandX ActiveX Command Utility keep these notes in mind:

- ❖ IdcCommandX must be initialized with a valid user and the `intradoc.cfg` directory. Outside of the `init` and `connection` managing methods, all methods use the serialized HDA format for communication.
- ❖ IdcCommandX attempts to establish a connection to a running server. If a connection is not made it fails.
- ❖ 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.



Tech Tip: See *Using the Stellent Launcher* (page 2-13) for information on using the Launcher (a native C++ application that allows a Java program to start as an NT service).

METHODS

init

```
Public Function init(UserName As String, StellentDir As String) As Boolean
```

Description

Initializes the system. This function initializes the system and must be called before using the IdcCommandX.

- ❖ The username and Stellent directory must be specified as parameters.
- ❖ Returns FALSE if the service fails to execute.

Required Parameters

This data is required:

Name	Description
UserName	A valid user name must be used.
StellentDir	The working directory. <ul style="list-style-type: none">❖ This is the directory containing the <i>intradoc.cfg</i> configuration file.❖ If this parameter is not set, the current working directory is used.

executeCommand

```
Public Sub executeCommand(Data As String)
```

Description

Executes a service from the content server. This method automatically handles the connection to the server. It will not establish a connection if a connection has already been established with a `connectToServer` call. On completion of the command, the connection is closed.

- ❖ Returns a string representing a serialized HDA file that holds the original request as well as the results.
- ❖ Returns the serialized *hda* containing `StatusCode` and `StatusMessage`.
- ❖ Returns `FALSE` if there is a connection failure.

Required Parameters

This data is required:

Name	Description
Data	A string representing a serialized HDA file. For example, the <code>IdcService</code> command and any necessary service parameters.

connectToServer

Public Function ConnectToServer() As Boolean

Description

Allows the calling program to establish a connection to the server. The connection is held open until a command is executed. Once a command is executed the connection is closed. This method does not need to be called. It is provided only as a convenience for managing the state of the connection.

- ❖ Returns TRUE on successful execution.
- ❖ Returns FALSE if there is a connection failure.

Required Parameters

No additional data is required.

closeServerConnection

Public Sub closeServerConnection()

Description

Closes the server connection. This method is provided as a convenience for managing the state of the connection. It does not need to be called after executing a command.

- ❖ Returns TRUE on successful execution.
- ❖ Returns FALSE if the service fails to execute.

Required Parameters

No additional data is required.

computeWebFilePath

Public Function computeWebFilePath(Data As String) As String

Description

Computes the web file path. Computes the web file path from the serialized *hda* and returns the value as a string.

- ❖ Returns the serialized *hda* containing *StatusCode* and *StatusMessage*.
- ❖ Returns the same data that was passed in as the parameter is returned with an additional value *WebFilePath* set in the local data containing the web file path.

Required Parameters

This data is required:

Name	Description
Data	A string representing a serialized HDA file.

computeNativeFilePath

Public Function computeNativeFilePath(Data As String) As String

Description

Computes the native file path. Computes the native file path from the serialized *hda* and returns the value as a string.

- ❖ Returns the serialized *hda* containing `StatusCode` and `StatusMessage`.
- ❖ Returns the same data that was passed in as the parameter is returned with an additional value `NativeFilePath` set in the local data containing the native file path.

Required Parameters

This data is required:

Name	Description
Data	A string representing a serialized HDA file.

computeURL

Public Function computeURL(Data As String, IsAbsolute As Boolean) As String

Description

Computes the URL path. Computes the URL from the serialized *hda* and returns the value as a string.

- ❖ Returns serialized *hda* containing *StatusCode* and *StatusMessage*.
- ❖ Returns the same data that was passed in as the parameter is returned with an additional value *URL* set in the local data containing the URL path.

Required Parameters

This data is required:

Name	Description
Data	A string representing a serialized HDA file.
IsAbsolute	Determine the absolute URL address.

USING SERVICES

INTRODUCTION

This chapter contains these topics:

- ❖ Conceptual Overview
- ❖ Service Structure
- ❖ Example Service: DOC_INFO
- ❖ Tutorial: Creating a Custom Service
- ❖ Tutorial: Creating a Query and Service Component
- ❖ Custom Application Example

CONCEPTUAL OVERVIEW

A IdcCommand service is a defined function or procedure that can be performed in the system. These services can be used by any program or HTML page to request information from the server or perform a specified function.

These services are discrete and may require parameters. They are the only way a client can talk to the server or access the database.

A service is the mechanism for interacting with the server and consequently the database. They are the only way user initiated functionality is implemented. A service call can be performed from either the client or server side and consists of a single or multiple actions along with a template page.

Service Request Process

This is the process when a user requests information on a particular content item from within a browser.

1. The user clicks the INFO button from the search results page.
2. The browser makes a request to the server using IdcService DOC_INFO.
3. The service performs an action consisting of a query to the database with dID as the parameter. This action is performed because the DOC_INFO service requires that dID (the unique content item ID) be passed as a parameter.
4. One of these responses occurs:
 - ❖ If the browser request did not contain dID, the content server reports a system error.
 - ❖ If the query returns no results, the user is informed that the content item is not in the system.
 - ❖ If the service locates the content item, it performs several other actions, which could be queries, or code, and finally merges the information that it has calculated and/or queried with an HTML template page.

Search Service Request Process

This is the process when a user places a search request from within a browser:

1. The user completes the search form and clicks the SEARCH button.
2. The browser sends a request to the web server to perform a search. When using the search engine, the URL for the request contains the following syntax:


```
/intradoc-cgi/idc_cgi_isapi.dll?
idcService=GET_SEARCH_RESULTS
```
3. The web server recognizes this request as a Stellent Content Server function and sends the specific request to the Stellent Content Server for processing. In doing so, the Content Server sends a request to the search engine using a search engine API.
4. The search engine sends the search results back to the Stellent Content Server. In turn, the results are sent to the web server. The web server then delivers the result of the search service to your web browser.

SERVICE STRUCTURE

The structure of a service-type resource is defined by a three column table. The table is delimited with a start tag `<@table "tablename"@>` and an end tag `<@end@>`. The first column contains the service's unique name. The second column describes the attributes of the service. The third column describes the actions that are performed by the service.

This example shows the HTML markup for a service entry in this table. This describes a service with *n* actions:

```
<tr>
<td>service name</td>
<td>service type
access level
```

Using Services

```
template page
sub-service
subjects notified<br>
error message</td>
<td>type of action1:function name1:function
parameters1:action control mask1:error message1[<br>]
type of action2:function name2:function parameters2:action
control mask2:error message2[<br>]
...
type of actionn:function namen:function parametersn:action
control maskn:error messagen</td>
</tr>
```

The `
` tag at the end of each action line is strictly for display purposes only and is optional. However, the `</td>` must occur on the same line as the last action.

Service Name

This column contains information about the unique name of the service.

```
<td>GET_DYNAMIC_PAGE</td>
```

The reference to a service called in a URL is the service name. For example, this URL is calling the service named `GET_DYNAMIC_PAGE`:

```
/intradoc-
cgi/idc_cgi_isapi.dll?IdcService=GET_DYNAMIC_PAGE&
PageName=index
```

Service Attributes

A service is defined by a script, which defines its name, attributes and actions. The attributes determine the type of service, the access level or security, template page, sub-service (or null), subjects notified, and an error message.

Each service consists of this structure:

- ❖ Service type
- ❖ Security or access level
- ❖ Template page
- ❖ Sub-Service (or null)
- ❖ Subjects notified
- ❖ Service error message
- ❖ Associated actions

The format is stored in an HTML table and a service consists of a single row in the table. On startup the content server parses the table and loads the services into memory.

The script format is as follows:

```
<tr>
<td>service name</td>
<td>service type
    access level
    template page
    sub-service
    subjects notified<br>
    error message</td>
```

Using Services

```
<td>type of action:function name:function
parameters:directives:error message
    type of action:function name:function
parameters:directives:error message
    ...</td>
</tr>
```

For example, the service for retrieving all documents checked out for a given user role is:

```
<tr>
  <td>CHECKIN_LIST</td>
  <td>DocService
    17
    CHECKIN_LIST
    null
    null<br>
    Unable to retrieve check in list.</td>
  <td>1:QcheckedoutDocs:CHECKIN_LIST:0:null
    3:computeDocInfoInHtmlPage:CHECKIN_LIST:0:Unable to
    compute user info.</td>
</tr>
```

Service Type

The service type defines the location of an HTM file containing the definition of service scripts. The type of service determines in part what actions can be called on it.

The types of services currently available are:

- ❖ Service—the default service.

- ❖ DocService—used for performing actions on documents. Examples are check in/out, document information, resubmit.
- ❖ FileService—used to retrieve files from the system.
- ❖ IndexerService—used to perform indexing related functions.
- ❖ MetaService—used to manage doc info fields.
- ❖ PageHandlerService—used by WebLayout Editor to edit pages.
- ❖ SearchService—used to perform search related functions.
- ❖ UserService—used to manage users (add/edit/delete).
- ❖ WorkflowService—used to manage workflows.
- ❖ WorkflowTemplateService—used to manage workflow templates.

Security or Access Level

Each service calls the global security check to determine if the logged on user has permission to execute the service. The global security check is only of relevance if the service requires global privilege. The check validates if the user needs to be part of the admin role or if he only needs a given privilege (less than ADMIN_PRIVILEGE) on at least one group.

The bit flags combined together to create an access level:

- ❖ READ_PRIVILEGE = 1
- ❖ WRITE_PRIVILEGE = 2
- ❖ DELETE_PRIVILEGE = 4
- ❖ ADMIN_PRIVILEGE = 8
- ❖ GLOBAL_PRIVILEGE = 16

For example, to access the administration page, the service requires the user to be part of the admin role. Consequently, he needs to have global admin privilege and the service has the access level set to 24. If on the other hand, the user wishes to access the check in page, he needs write privilege on at least one group, and the access level of the security group is set to 18.

The GLOBAL_PRIVILEGE flag is used to for these two purposes:

- ❖ If asking for admin privilege, ensures that the user has some type of admin or subadmin rights.
- ❖ Otherwise makes sure that the user has the requested access level to at least one security group in the content server. This is only relevant if the "guest" role has no rights at all and you want to force a user to login when they access such pages as the "Query Page" (note the Query Page uses the GLOBAL_PRIVILEGE flag through the service GET_DOC_PAGE).

Template page

The template page is used to communicate back to the browser the successful request. Information that the service gathers is merged with the template page. Not all types of services require or even use a template page, e.g. the PageHandlerService. The template page name is mapped to an HTML file listed in the file ~/resources/templates/template.hda.

Sub-Service

The service may define a sub-service to execute, otherwise, the value null is used. For example, the service ADD_WORKFLOWDOCUMENT executes the sub-service ADD_WORKFLOWDOCUMENT_SUB. This sub-service is a workflow related service that adds a revised content item to the workflow and consists of these actions:

- ❖ Queries whether the content item workflow is locked.
- ❖ Inserts the workflow content item information in the database.
- ❖ Retrieves the workflow content item name from the database.

- ❖ Evaluates the revision status of the content item.
- ❖ Creates a new revision.

Subjects Notified

If a service changes one or more subjects, it must notify others of these changes. The subjects notified string is a comma separated lists of changed subjects. For example, the `ADD_USER` service adds a new user to the system, and subsequently informs the system that the “users” subject has changed. Possible subjects are: aliases, doctypes, docformats, documents, dynamicqueries, metadata, metaoptlists, pagelist, reports, templates, users, and workflows.

Service Error Message

The error message returned by the service, if no action has overridden it. Each action has an error message associated with it. If this message is not null, it becomes the error message for the remainder of the actions in the service. If it is null, the error message remains unchanged from the previous action. For example, the error message of `CHECKIN_LIST` is "Unable to retrieve check in list" but on executing the second action it becomes "Unable to compute allowable user actions."

Associated Actions

Each service may contain one or more actions, which determines what happens on execution. Each service by default will inform its requestor of changes to subjects. Consequently, the `PING_SERVER` service, which has no actions, is used by applets to detect changes in the state of the server.

Action Type

An action can be used to execute an SQL statement, perform a query, do code, cache the results of a query, and load an option list. The possible types of actions are:

- ❖ QUERY_TYPE = 1
- ❖ EXECUTE_TYPE = 2
- ❖ CODE_TYPE = 3
- ❖ OPTION_TYPE = 4
- ❖ CACHE_RESULT_TYPE = 5

Action Function

- ❖ The function is restricted by the type of service and the type of action.
- ❖ For QUERY_TYPE, the function must be a “select” query.
- ❖ For EXECUTE_TYPE, the function specifies a query that performs an action on the database.
- ❖ For CODE_TYPE, the function specifies a code module that is a part of the java class implementing the service.
- ❖ For OPTION_TYPE, the function refers to an option list stored in the system.
- ❖ For CACHE_RESULT_TYPE, the function is as in QUERY_TYPE, but here the results returned by the query are stored for later use.

The difference between QUERY_TYPE and CACHE_RESULT_TYPE is that in the first case the query is immediately discarded as soon as another query is executed.

Action Parameters

The parameters are a comma-separated that are used by the functions. In the case of QUERY_TYPE and CACHE_RESULT_TYPE, the first parameter will be the name the action assigns to the result set returned from the query.

For OPTION_TYPE, the parameters are optional, but if they are given they are used as follows: the first parameter is the key under which the option list is loaded, the second parameter is the selected value for display on an HTML page.

Action Control Mask

The control mask is especially useful in controlling the results from queries to the database. Possible bit values and there meanings:

- ❖ CONTROL_IGNORE_ERROR = 1 - Do not abort the service on error
- ❖ CONTROL_MUST_EXIST = 2 - At least one record must be returned by the query.
- ❖ CONTROL_BEGIN_TRAN = 4 - Starts a database transaction
- ❖ CONTROL_COMMIT_TRAN = 8 - Concludes a database transaction
- ❖ CONTROL_MUST_NOT_EXIST = 16 - Query must not return any rows

CONTROL_MUST_EXIST and CONTROL_MUST_NOT_EXIST are used only for QUERY_TYPE and CACHE_RESULT_TYPE.

Action Error Message

Each action has an error message associated with it. If this message is not null, it becomes the error message for the remainder of the actions in the service. If it is null, the error message remains unchanged from the previous action. For example, the error message of CHECKIN_LIST is “Unable to retrieve check in list” but on executing the second action it becomes “Unable to compute allowable user actions.”

EXAMPLE SERVICE: DOC_INFO

This section describes the DOC_INFO service and provides code samples from the std_services HTM file and the doc_info HTM template file.

These are the defined attributes of the DOC_INFO service.

- ❖ Service Type: DocService—This service is providing information about a content item.
- ❖ Access Level: 1—The user requesting the service must have read privilege on the content item.
- ❖ Template Page: DOC_INFO—This service uses the DOC_INFO template (doc_info.htm file).
- ❖ Sub Service: null—This service does not define a sub-service to execute.
- ❖ Subjects Notified: null—No subjects are affected by this service.
- ❖ Error Message: Unable to retrieve information about the revision.

Service Definition File

The DOC_INFO service definition from the `<home>/config/resources/std_services.htm` file is shown:

```
<tr>
<td>DOC_INFO</td>
<td>DocService
1
DOC_INFO
null
null<br>
Unable to retrieve information about the revision.</td>
```

```

<td>5:QdocInfo:DOC_INFO:2: This document no longer exists.
3:checkSecurity:DOC_INFO:0:Unable to retrieve information
for ''{dDocName}'' .
3:getDocFormats:QdocFormats:0:null
3:getURLAbsolute::0:null
3:getUserMailAddress:dDocAuthor,AuthorAddress:0:null

3:getUserMailAddress:dCheckoutUser,CheckoutUserAddress:0:nu
ll
3:getWorkflowInfo:WF_INFO:0:null
3:getDocSubscriptionInfo:QisSubscribed:0:null
5:QrevHistory:REVISION_HISTORY:0: Unable to retrieve
revision history for ''{dDocName}''.<br></td>
</tr>

```

Service Template File

The template page for the DOC_INFO service is the DOC_INFO template. It is important to know what is happening between the files so that you can understand the interactions between the template page and the actions performed in a service.

The definition for the content that the doc_info.htm template contains is located in the <home>/shared/config/resources/std_page.htm file. Code from both files appear in the following markup section:

Markup from the <home>/shared/config/templates/doc_info.htm file:

```

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>
<head>
<$include std_info_html_head_declarations$>

```

Using Services

```
</head>
<$include info_body_def$>
<$include info_page_content$>
</body>
</html>
```

Markup from the `<home>/shared/config/resources/std_page.htm` file that defines what will appear in the `doc_info.htm` template:

```
<@dynamichtml info_page_content@>
<$include std_page_begin$>
<$include std_header$>
...
<!-- Do a loop on DOC_INFO so that all substitution tags
will use DOC_INFO as their first place to find their
values. Otherwise there is confusion between this result
set and the REVISION_HISTORY table that comes later. For
example 'dStatus' is a value in both tables-->
<$loop DOC_INFO$>
<$if AllowPrimaryMetaFile and isTrue(AllowPrimaryMetaFile)
and
isTrue(dFormat like "*idcmeta*")$>
<$showPrimaryMetaFileFields = "1"$>
<$endif$>
<$include doc_info_notify_data$>

<table border=0 cellpadding=2 cellspacing=0
width=<$docInfoWidth-30$>>
<caption align=top><h4
class=pageTitle><$pageTitle$></caption>
```

```

<$include special_checkin_fields1$>
<$include std_revision_label_field$>
<$include std_document_type_field$>
<$include std_document_title_field$>
<$include author_checkin_field$>
<$include std_meta_fields$>
<$include security_checkin_fields$>
<$include checkout_author_info_field$>
<$if IsStagingDoc$>
<$include doc_date_fields$>
<$endif$>
<$fieldName = "dStatus", fieldCaption = "Status"$><$include
std_displayonly_field$>
<$if HasOriginal$>
<$fieldName = "dDocFormats", fieldCaption =
"Formats"$><$include std_display_field$>
<$endif$>
<$include workflow_list_for_doc$>
<$if HasUrl$>
<$include doc_url_field$>
<$endif$>
<$if HasOriginal and not ClientControlled and not
showPrimaryMetaFileFields$>
<$fieldName = "dOriginalName", fieldCaption = "Get Native
File"$>
<$if DownloadApplet$>
<$valueStyle="xxsmall", fieldValue =
strTrimWs(inc("download_file_by_applet_form_content"))$>

```

Using Services

```
<$else$>
<$fieldValue = strTrimWs(inc("doc_file_get_copy"))$>
<$endif$>
<$if DownloadApplet$><form name=downloadForm><$endif$>
<$include std_displayonly_field$>
<$if DownloadApplet$></form><$endif$>
<$endif$>
<$if IsFailedConversion or IsFailedIndex or
IsDocRefinePassthru$>
<$if IsFailedConversion$><$include
std_namevalue_separator$><$endif$>
<tr>
<td align=right><span class=errorHighlight>
<$if IsFailedIndex$>Index Error:
<$else$>Conversion Error:
<$endif$></span></td>
<td>
<table>
<tr>
<td><span class=tableEntry>
<$dMessage$>
<$if IsFailedIndex$>
<br>Content has been indexed with Info only.
Resubmit should only be performed if the problem has been
resolved.
<$elseif IsDocRefinePassthru$>
<br>Content Refinery failed to convert the content item but
released it to the
```

```

web by copying the native file.
<$endif$></span></td>
<td><form action="<$HttpCgiPath$>" method="POST">
<input type=hidden name=dID value="<$dID$>">
<input type=hidden name=dDocName value="<$dDocName$>">
<input type=hidden name=IdcService
value="RESUBMIT_FOR_CONVERSION">
<input type=submit value=" Resubmit ">
<$if ClientControlled$>
<input type=hidden name=ClientControlled value="DocMan">
<$endif$>
</form></td>
</tr>
</table>
</td>
</tr>
<$if IsFailedConversion$><$include
std_namevalue_separator$><$endif$>
<$endif$>
</table>
<$if IsNotSyncRev$>
<table width="100%">
<tr>
<td align=center><span class=errorHighlight>The local copy
of this content item has
not been updated to the latest revision. Use <i>Get Native
File</i> or <i>Check out</i>

```

Using Services

```
to update your local copy of
<i><$dDocName$></i>.</span></td>
</tr>
</table>
<$endif$>

<$if IsStagingDoc$>
<br>
<table width="90%">
<tr>
<td width="20%" align=center><$include
doc_problem_reports$></td>
<td width="20%" align=center><$include
project_problem_reports$></td>
</tr>
</table>
<$include doc_provider_info$>
<$else$>
<table width="90%">
<tr>
<$if ClientControlled$>
<td width="20%" align=center><$include
doc_select_actions$></td>
<$else$>
<td width="20%" align=center><$include
doc_file_undo_checkout$></td>
<td width="20%" align=center><$include
doc_file_checkout$></td>
```

```

<td width="20%" align=center><$if
showPrimaryMetaFileFields$><$include meta_file_update$>
<$else$><$include doc_file_update$><$endif$></td>
<$endif$>
<td width="20%" align=left><$include
doc_subscription_unsubscription$></td>
<$if ClientControlled$>
<td width="20%"></td>
<td width="20%"></td>
<$endif$>
</tr>
</table>
<$endif$>
<$if HasOriginal and DownloadApplet$>
<$include download_native_applet$>
<$endif$>

<!-- end loop on DOC_INFO-->
<$endloop$>
<$if IsStagingDoc$>
<!-- present a problem report form -->
<$include doc_add_problem_report$>
<$else$>
<!-- Table holding information about all revisions of this
document-->
<$include doc_rev_table$>
<$endif$>
</td>

```

```
</tr>  
</table>  
<$include std_page_end$>  
<@end@>
```

Service Actions

The DOC_INFO service consists of these actions:

Action 1— Cached query action that retrieves information from the database using a query. This action retrieves content item information. The result of this query is assigned to the parameter DOC_INFO and stored for later use. The control mask setting specifies that the query must return a record or the action fails with the given error message. The action throws a data exception if the content item no longer exists and returns this error message: *This content item no longer exists.*

Action 2— Code action specifying a code module that is a part of the Java class implementing the service. This action retrieves the data assigned to the parameter DOC_INFO and maps the result set values for *dStatus* and *dDocTitle*.

Action 3—Code action specifying a code module that is a part of the Java class implementing the service. This action retrieves the data assigned to the parameter DOC_INFO and evaluates the assigned security level to verify that the user is authorized to perform this action. If the user fails the security check this error message is returned: *Unable to retrieve information for "{dDocName}"*.

Action 4—Code action specifying a code module that is a part of the Java class implementing the service. This action retrieves the file formats for the content item. The action passes *QdocFormats* as a parameter (defined in `<install_dir>/config/resources/query.htm`). The file formats are passed to the *Formats*: entry of the DOC_INFO template.

Action 5—Code action specifying a code module that is a part of the Java class implementing the service. This action resolves the URL of the content item. The URL is passed to the *Web Location*: entry of the DOC_INFO template.

Action 6—Code action specifying a code module that is a part of the Java class implementing the service. This action resolves the email address of the content item author and the user who has checked out the content item. The action passes *dDocAuthor* and *AuthorAddress* as parameters.

Action 7—Code action specifying a code module that is a part of the Java class implementing the service. This action resolves the email address of the content item author and the user who has checked out the content item. The action passes *dCheckoutUser* and *CheckoutUserAddress* as parameters.

Action 8—Code action specifying a code module that is a part of the Java class implementing the service. This action evaluates whether the content item is part of a workflow. The action passes WF_INFO as a parameter. The DOC_INFO template is referenced and if WF_INFO exists then workflow information is included in the DOC_INFO template.

Action 9—Code action specifying a code module that is a part of the Java class implementing the service. This action evaluates whether the current user has subscribed to the content item and modifies the DOC_INFO page. If the current user is subscribed, an **Unsubscribe** button is displayed. If the user is not subscribed, a **Subscribe** button is displayed. The action passes *QisSubscribed* as a parameter (defined in `<install_dir>/config/resources/query.htm`).

Action 10—Cached query action that retrieves information from the database using a query. This action retrieves revision history information. The result of this query is assigned to the parameter REVISION_HISTORY. The DOC_INFO template uses REVISION_HISTORY in a loop to present information about each revision in the DOC_INFO page. If the action fails, the service returns this error message: *Unable to retrieve revision history for "{dDocName}"*.

TUTORIAL: CREATING A CUSTOM SERVICE

Overview

These are the steps required to create a new service:

1. Define a service in an HTM file. The file must include a table that is identical in structure to the StandardServices table.
2. Load your service by defining it in a ResourceDefinition ResultSet.
3. Merge the table defining your service with the StandardServices table.

Steps

Follow these steps the create a custom service:

1. Make a copy of the file <home>/shared/config/resources/std_services.htm and place it into your component's resources/ directory.
2. Edit the table definition column entries for Name, Attributes, and Actions.

This is an example of a defined custom service named My Services and the coding involved:

```
<HTML>
<HEAD>
<META HTTP-EQUIV='Content-Type' content='text/html;
charset=iso-8859-1'>
<TITLE>Custom Scripted Services</TITLE>
</HEAD>
<BODY>
<@table MyServices@>
```

```

<table border=1><caption><strong>Scripts For Custom
Services
</strong></caption>
<tr>
<td>Name</td><td>Attributes</td><td>Actions</td>
</tr>
<tr>
<td>ADD_REPORT</td>
<td>Service
18
ADD_REPORT_FORM
null
null<br>
Unable to add report.</td>
<td>2:Ireport:0:null</td>
</tr>
<tr>
<td>REPORTS_LIST</td>
<td>Service
17
REPORT_LIST_FORM
null
null<br>
Unable to retrieve reports.</td>
<td>5:Qreports:REPORT_LIST:0:null</td>
</tr>
</table>

```

Using Services

```
<@end@>  
<br><br>  
</BODY>  
</HTML>
```

3. Create a reference to your custom file in your ResourceDefinition ResultSet
4. Create a MergeRule that merges MyServices with the Services table.

The accompanying resource definition file would look similar to the following:

```
@ResultSet ResourceDefinition  
4  
type  
filename  
tables  
loadOrder  
service  
resources/MyServices.htm  
MyServices  
1  
@end  
@ResultSet MergeRules  
3  
fromTable  
toTable  
column  
MyServices  
Services  
name  
@end
```

TUTORIAL: CREATING A QUERY AND SERVICE COMPONENT

Overview

In this section, you will create a new query to read from the subscriptions table, a new service to call the query, and a new template page to display the results. A button will be added to the content information page to call the new service. The button on the Information page will be displayed next to the Subscribe/Unsubscribe button.

We will use the Component Wizard to create a new project. We will then create a new template page that will display the subscription information. Next, we will create the query to retrieve subscriptions from the database and then we will create the service, which will call that query. Finally, we will update a resource to add our new button to the doc info page.

Steps

Follow these step to create the new service component:

1. Open the Component Wizard and create a new component named SubscribedUsers.
2. Add a new Template selecting the error page as the base to begin from. Enter these parameters as shown:

Name: DOC_SUBSCRIBERS

Class: Message

Form Type: SubscriptionList

File Name: SubscribedUsers_List.htm

Description: Users subscribed to document page

3. Edit the template file as shown:

```
<html>
<head>
<$defaultPageTitle="Subscribed Users"$>
<$include std_html_head_declarations$>
</head>
<$include body_def$>
<$include std_page_begin$>
<$include std_header$>
<!--Subscriber List-->
<table border="0" cellpadding="0" cellspacing="0">
<tr>
<td width="465" align="left">
<table border = "0" width="465" align="left"
valign="middle">
<tr><td ><span class=errorMessage>
Users Subscribed to <$dDocTitle$>
</span></td></tr>
<tr><td>
<$loop UserList$>
</td></tr>
<tr><td>
<$dSubscriptionAlias$>
<$endloop$>
</td>
</tr>
</table>
```

```

</td>
</tr>
</table>
<$include std_page_end$>
</body>
</html>

```

Add a query to your component

1. Click **Add** on the lower left of the Component Wizard.
2. Select Query and ensure that these entries are correctly defined:

File Name: resources/subscribedusers_query.htm

Load Order: 1

3. Click **Next**.
4. Ensure that this entry is correctly defined:

Table Name: SubscribedUsers_Queries

5. Click **Next**.

6. Enter the query information as shown:

Name: QusersSubscribedCustom

Query: SELECT *FROM Subscription
WHERE dSubscriptionID = ?

7. Ensure that these Parameter entries are correctly defined:

Name: dsubscription

Type: varchar

8. Click Finish. The query is complete.

Add a Service

1. Click **Add** on the lower left of the Component Wizard.
2. Select Service and click **Next**.
3. Ensure that this entry is correctly defined:
Table Name: SubscribedUsers_Services
4. Click **Next**.
5. Enter the service information as shown:
Name: GET_SUBSCRIBERS
Service Class: DocService
Template: DOC_SUBSCRIBERS
Access Level: Read
Error Message: Unable to retrieve subscriptions for “{dDocName}”.
6. Click **Add** to add the query.
7. Enter the Add Action information as shown. The Action is the query that we just created and the Parameter is the name of the ResultSet that is accessed on the template page.
Action: QusersSubscribedCustom
Parameters: UserList
8. Click **OK**.
9. Click **Finish** on the Add Service screen. The new service is complete.

Create the button that will link to this new service and page

1. Add a new resource to the component by clicking the Add button on your Component Wizard then selecting Resource - HTML Include.
2. Click the **Next** button then Select the doc_subscription_unsubscription include. Delete all the code and replace it with the following:

```
<${include super.doc_subscription_unsubscription$}>
<td>
<form name=Subscribers action="<HttpCgiPath$>"
method="GET">
<input type=hidden name=IdcService
value="GET_SUBSCRIBERS">
<input type=hidden name=dSubscriptionId
value="<${dDocName$}>">
<input type=hidden name=dDocTitle value="<${dDocTitle$}>">
<input type=hidden name="Users Subscribed">
</form>
</td>
```

3. Your component is now complete and ready to enable and test. Remember to restart your server once you have enabled it.
4. Test the component by performing a search and clicking on the info button from the search results page. You will need to subscribe to the document before any subscriptions are available to be viewed on the page.

CUSTOM APPLICATION EXAMPLE

This example application call five services and defines six private functions.

Services Called

These services are called and a serialized HDA string is built for each:

- ❖ CHECKOUT_BY_NAME
- ❖ CHECKIN_UNIVERSAL
- ❖ DOC_INFO
- ❖ GET_FILE

These parameters for CHECKIN_UNIVERSAL are defined:

- ❖ doFileCopy
- ❖ dDocName
- ❖ dDocTitle
- ❖ dDocType
- ❖ dSecurityGroup
- ❖ dDocAuthor
- ❖ dDocAccount
- ❖ primaryFile

The GET_TABLE service is called and this parameter is defined:

- ❖ tableName

Private Functions

These private functions are defined:

- ❖ getNativeFilePath
- ❖ checkOutByName
- ❖ checkinUniversal
- ❖ getDocInfo
- ❖ getFile
- ❖ parseResultSet

Sample Code

```
' Defines a private function.
Private Function getNativeFilePath() As String
Dim idccmd As IdcCommandX
Dim str As String
Dim res
Dim dID As String, dExtension As String, dDocType As
String, dDocAccount As String

dID = "2"
dExtension = "pdf"
dDocType = "acc"
dDocAccount = ""

' Builds a serialized HDA string.
str = "@Properties LocalData" + vbCrLf
str = str + "dID=" + dID + vbCrLf
str = str + "dExtension=" + dExtension + vbCrLf
str = str + "dDocType=" + dDocType + vbCrLf
str = str + "dDocAccount=" + dDocAccount + vbCrLf
str = str + "@end" + vbCrLf

Set idccmd = New IdcCommandX
res = idccmd.init("sysadmin", "c:\stellent\bin")
res = idccmd.computeNativeFilePath(str)
```

Using Services

```
Open "c:\newdoc.txt" For Binary Access Write As #1
Put #1, , str
Put #1, , res
Close #1
MsgBox (res)
End Function

' Defines a private function.
Private Function checkOutByName(ByVal dDocName As String)
As String
Dim idccmd As IdcCommandX
Dim idcService As String, str As String
Dim res

' Calls a service and builds a serialized HDA string.
idcService = "CHECKOUT_BY_NAME"
str = "@Properties LocalData" + vbCrLf
str = str + "IdcService=" + idcService + vbCrLf
str = str + "dDocName=" + dDocName + vbCrLf
str = str + "@end" + vbCrLf

' In an actual application the return codes need to be
handled. For this example, the service is called while
there is no content with that specific dDocName.
Set idccmd = New IdcCommandX
res = idccmd.init("sysadmin", "c:\stellent\bin")
res = idccmd.executeCommand(str)
End Function
```

```

' Defines a private function.
Private Function checkinUniversal(ByVal doFileCopy As
String, ByVal dDocName As String, ByVal dDocTitle As
String, ByVal dDocType As String, ByVal dSecurityGroup As
String, ByVal dDocAuthor As String, ByVal dDocAccount As
String, ByVal primaryFile As String) As String

' Builds a serialized HDA string.
Dim idccmd As IdcCommandX
Dim idcService, res, str As String

' Calls a service and builds a serialized HDA string.
idcService = "CHECKIN_UNIVERSAL"
str = "@Properties LocalData" + vbCrLf
str = str + "IdcService=" + idcService + vbCrLf
str = str + "doFileCopy=" + doFileCopy + vbCrLf
str = str + "dDocName=" + dDocName + vbCrLf
str = str + "dDocTitle=" + dDocTitle + vbCrLf
str = str + "dDocType=" + dDocType + vbCrLf
str = str + "dSecurityGroup=" + dSecurityGroup + vbCrLf
str = str + "dDocAuthor=" + dDocAuthor + vbCrLf
str = str + "dDocAccount=" + dDocAccount + vbCrLf
str = str + "primaryFile=" + primaryFile + vbCrLf
str = str + "@end" + vbCrLf

' exec hda...
Set idccmd = New IdcCommandX

```

Using Services

```
res = idccmd.init("sysadmin", "c:\stellent\bin")
res = idccmd.executeCommand(str)
MsgBox (CStr(res))
End Function

' Defines a private function.
Private Function getDocInfo(ByVal dID As String) As String
Dim idccmd As IdcCommandX
Dim idcService As String, str As String
Dim res

' Calls a service and builds a serialized HDA string.
idcService = "DOC_INFO"
str = "@Properties LocalData" + vbCrLf
str = str + "IdcService=" + idcService + vbCrLf
str = str + "dID=" + dID + vbCrLf
str = str + "@end" + vbCrLf

' exec hda....
Set idccmd = New IdcCommandX
res = idccmd.init("sysadmin", "c:\stellent\bin")
res = idccmd.executeCommand(str)
MsgBox (res)
End Function

' Defines a private function.
```

```

Private Function getFile(ByVal dID As String, ByVal
dDocName As String, ByVal RevisionSelectionMethod As
String, ByVal Rendition As String)
Dim idccmd As IdcCommandX
Dim idcService, str As String

Dim res As Variant
Dim fileName As String
Dim fileSize As Long
Dim indexStop As Integer

' Calls a service and builds a serialized HDA string.
idcService = "GET_FILE"
str = "@Properties LocalData" + vbCrLf
str = str + "IdcService=" + idcService + vbCrLf
str = str + "dDocName=" + dDocName + vbCrLf
str = str + "dID=" + dID + vbCrLf
If (RevisionSelectionMethod = "Specific" Or
RevisionSelectionMethod = "Latest" Or
RevisionSelectionMethod = "LatestReleased") Then

' Ignore dDocName and use dID instead.
str = str + "RevisionSelectionMethod=" +
RevisionSelectionMethod + vbCrLf
End If

If (Revision = "Primary" Or Revision = "Web" Or Revision =
"Alternate") Then
str = str + "Revision=" + Revision + vbCrLf

```

Using Services

```
End If
str = str + "@end" + vbCrLf

' exec hda...
Set idccmd = New IdcCommandX
res = idccmd.init("sysadmin", "c:\stellent\bin")
res = idccmd.executeCommand(str)

Open "c:\newdoc.txt" For Binary Access Write As #1
Put #1, , res
Close #1
MsgBox (Len(res))

' chop at filename= and store fileName
indexStop = InStr(res, "filename=")
tmpStr = (Mid(res, indexStop))
indexStop = InStr(tmpStr, Chr(13))
fileName = Mid(tmpStr, 10, indexStop - 10)
MsgBox (fileName)
' MsgBox (CStr(Asc(Mid(tmpStr, 2, 1))))

' chop at Content-length: and store fileSize
tmpStr = Mid(tmpStr, indexStop)
indexStop = InStr(tmpStr, "Content-Length: ")
tmpStr = (Mid(tmpStr, indexStop))
indexStop = InStr(tmpStr, Chr(10))
fileSize = CLng(Mid(tmpStr, 17, indexStop - 17))
```

```
MsgBox (CStr(fileSize))
MsgBox (Len(res))
End Function

Private Sub cmdAddUser_Click()
frmAddUser.Show
End Sub

Private Sub cmdCheckin_Click()
Dim idcService As String, doFileCopy As String, dDocName As
String
Dim dDocTitle As String, dDocType As String, dSecurityGroup
As String
Dim dDocAuthor As String, dDocAccount As String, primayFile
As String

' Calls a service and defined parameters.
idcService = "CHECKIN_UNIVERSAL"
doFileCopy = "1"
dDocName = "myDocNameNewh"
dDocTitle = "myDocTitleb"
dDocType = "ADACCT"
dSecurityGroup = "Public"
dDocAuthor = "Jennifer"
dDocAccount = ""
primaryFile = "c:/junk_b.doc"

' In an actual application check for errors.
```

Using Services

```
' Lock the file in order to upload a new revision.
Call checkOutByName(CStr(dDocName))
' If the dDocName is not in the system, it gets added as
first revision
' by the CHECKIN_UNIVERSAL call.
Call checkinUniversal(doFileCopy, dDocName, dDocTitle,
dDocType, dSecurityGroup, dDocAuthor, dDocAccount,
primaryFile)
End Sub

Private Sub cmdDocInfo_Click()
Call getDocInfo("269")
End Sub

Private Sub cmdDownload_Click()
Call ba
End Sub

Private Sub cmdGetFile_Click()
Call getFile("14", "", "", "")
End Sub

Private Sub cmdGetNativeFile_Click()
Call getNativeFilePath
End Sub

Private Sub Command1_Click()
Dim idccmd As IdcCommandX
```

```

Dim res, str
Open "c:\adduser.txt" For Append As #1
Set idccmd = New IdcCommandX
res = idccmd.init("sysadmin", "c:\stellent\bin")
str = "@Properties LocalData" + vbCrLf +
"IdcService=ADD_USER" + vbCrLf + "dName=Jennifer" + vbCrLf +
"dFullName=Jennifer Smith" + vbCrLf +
"dPassword=password" + vbCrLf + "dEmail=email@email.com" +
vbCrLf + "dUserAuthType=LOCAL" + vbCrLf + "@end" + vbCrLf +
"@ResultSet UserAttribInfo" + vbCrLf + "2" + vbCrLf +
"dUserName" + vbCrLf + "AttributeInfo" + vbCrLf +
"Jennifer" + vbCrLf + "role,admin,15" + vbCrLf + "@end" +
vbCrLf
res = idccmd.executeCommand(str)
Print #1, res
Close #1
End Sub

Private Sub Command2_Click()
Dim idccmd As IdcCommandX
Dim res, str

Dim myRS As String
Dim idcService, tableName As String

idcService = "GET_TABLE"
tableName = "Accounts"

Open "c:\a_getsec.txt" For Append As #1

```

Using Services

```
Set idccmd = New IdcCommandX
res = idccmd.init("sysadmin", "c:\stellent\bin")
str = "@Properties LocalData" + vbCrLf + "IdcService=" +
idcService + vbCrLf + "tableName=" + tableName + vbCrLf +
"@end" + vbCrLf
myRS = idccmd.executeCommand(str)

' Parse out the results set.
Call parseResultSet(myRS, tableName)
Print #1, res
Close #1
End Sub

' Calls a private function.
Private Function parseResultSet(strResultsSet As String,
strSearchString As String) As String
Dim indexStop As Integer
Dim tmpStr As String

Dim numberOfRows As Integer
Dim numberOfElementsInSet As Integer
Dim resultElement()
MsgBox (strResultsSet)

' Start of results set.
indexStop = InStr(strResultsSet, "@ResultSet " &
strSearchString)
```

```

' Check for error (0 index) before moving on.
tmpStr = (Mid(strResultsSet, indexStop))
MsgBox (tmpStr)

' Determine how many data lines are in the HTA file.
indexStop = InStr(tmpStr, "@end")
For i = 1 To indexStop
If (Mid(tmpStr, i, 1) = Chr(10)) Then
numberOfRows = numberOfRows + 1
End If
Next i
numberOfRows = numberOfRows - 2 ' Remove the first line of
data.

' Find first line that identifies the ResultsSet
indexStop = InStr(tmpStr, Chr(10))
tmpStr = (Mid(tmpStr, indexStop + 1))

' Get number of elements in record set, chop the line off
the record set...
indexStop = InStr(tmpStr, Chr(10))
numberOfElementsInSet = CInt((Mid(tmpStr, 1, indexStop)))
tmpStr = (Mid(tmpStr, indexStop + 1))

' Set storage array.
ReDim resultElement((numberOfRows / numberOfElementsInSet),
numberOfElementsInSet)
Dim junk As String

```

Using Services

```
' Populate array from HTA dataset
For i = 1 To (numberOfRows / numberOfElementsInSet)
For j = 1 To numberOfElementsInSet
indexStop = InStr(tmpStr, Chr(10))
resultElement(i, j) = Mid(tmpStr, 1, indexStop - 1)
tmpStr = (Mid(tmpStr, indexStop + 1))
junk = junk + resultElement(i, j)
Next j
Next i
MsgBox (junk)
parseResultSet = "je"
End Function

' Set storage array.
Sub ba()
Dim b() As Byte 'This byte array will capture the file
Dim strURL As String ' URL string
Dim strDest As String ' Destination File

' Set the strURL to a valid address.
'strURL = "http://localhost/intradoc-
cgi/idc_cgi_isapi.dll?IdcService=GET_FILE&dID=14"
'strDest = "C:\myjunk.html"
strURL = "localhost"
b() = Inet1.OpenURL(strURL, icByteArray)
'Open strDest For Binary Access Write As #1
'Put #1, , b()
```

```
'Close #1  
End Sub
```


Chapter

5

IDCCOMMAND SERVICES

INTRODUCTION

This chapter contains these topics:

- ❖ Conceptual Overview
- ❖ Available Services

CONCEPTUAL OVERVIEW

These commands can be called from the IdcCommand and IdcCommandX interface. Each scripted service defined in the content server can be used as a command. The service can have required and/or optional parameters.

- ❖ Execute a PING_SERVER request before calling other services to ensure that there is a connection to the content server and that you are logged in as a user authorized to execute commands (see page 5-298).
- ❖ All commands must have the IdcService parameter set to the service that is to be executed.

Admin Server Related Services

These services are used by the Admin Server to request information (see Available Services starting on page 5-8 for additional information on each of these services):

Service	Description
CONFIG_INFO	This service retrieves configuration information for the currently active component.
DOWNLOAD_COMPONENT	This service executes a download of the existing component.
GET_SERVER_OUTPUT	This service loads the server output for display.
UPLOAD_NEW_COMPONENT	This service unpackages the files according to the manifest and uploads the new component for installation.

Content Publisher Related Services

These services are used by Content Publisher (see Available Services starting on page 5-8 for additional information on each of these services):

Service	Description
DOC_INFO_LATESTRELEASE	This service retrieves the latest revision.
GET_DOC_CONFIG_INFO	This service retrieves configuration information.
GET_DOCUMENT_PROBLEMREPORTS	This service provides a content item problem report.

Customization Related Services

These services are frequently used for customization (see Available Services starting on page 5-8 for additional information on each of these services):

Service	Description
ADD_USER	This service adds a new user to the system.
CHECKIN_UNIVERSAL	This service performs a content server controlled check in.
CHECKOUT	This service checkouts the latest revision of the specified content item.
CHECKOUT_BY_NAME	This service checks out the latest revision of the specified content item.
DELETE_USER	This service deletes an existing user.
EDIT_USER	This service edits an existing user.

Service	Description
GET_DYNAMIC_CONVERSION	This service returns the latest revision, or the latest released revision of a content item.
GET_FILE	This service returns a specific rendition of a content item, the latest revision, or the latest released revision. A copy of the file is retrieved without performing a check out.
GET_TABLE	This service exports the specified table in the content server database.
MERGE_TABLE	This service allows the user to add rows to any table in the content server database.
PAGE_HANDLER	This service rebuilds the static web layout structure as it is defined in the Web Layout Editor.
START_SEARCH_INDEX	This service either updates or rebuilds the search index.
UNDO_CHECKOUT	This service reverses a content item checkout.
UNDO_CHECKOUT_BY_NAME	This service reverses a content item checkout using the Content ID.
UPDATE_META_TABLE	This service updates the DocMeta table with the newly added or deleted custom metadata columns.
UPDATE_OPTION_LIST	This service adds or updates the specified option list.

Personalization Related Services

These services are used for personalization (see Available Services starting on page 5-8 for additional information on each of these services):

Service	Description
GET_PORTAL_PAGE	This service loads configuration information for the template page, retrieves the option lists for custom fields, and dynamically displays the portal page.
GET_SELF_REGISTER_PAGE	This service retrieves the self registration page and loads the user metadata information for user self registration
PNE_GET_SEARCH_RESULTS	This service returns the search results to the personal navigation environment for display to the user.
PNE_SAVE_QUERY	This service saves the query results to the personal navigation environment for display to the user.
PNE_UPDATE_PERSONAL_URLS	This service updates the personal URLs of the personal navigation environment for display to the user.
PNE_UPDATE_PORTAL_INFO	This service updates the personal URLs of the personal navigation environment for display to the user.

Problem Report Related Services

These services relate to problem reports (see Available Services starting on 5-8 for additional information on each of these services):

Service	Description
ADD_PROBLEMREPORT	This workflow related service adds a problem report to the content item.
DELETE_PROBLEMREPORT	This workflow related service deletes a problem report.
GET_DOCUMENT_PROBLEMREPORTS	This content publisher related service provides a content item problem report.
GET_UPDATE_PROBLEMREPORT_FORM	This workflow related service retrieves the problem report update form.
PROBLEMREPORT_INFO	This workflow related service returns problem report information.
RESEND_PROBLEMREPORT	This workflow related service resends a problem report.
UPDATE_PROBLEMREPORT	This workflow related service updates the problem report.

Provider Related Services

A provider is an Application Programming Interface (API) that establishes a connection to outside entities such as databases, LDAP servers, server sockets, or other Content Server instances.

These services are provider related (see Available Services starting on 5-8 for additional information on each of these services):

Service	Description
ADD_EDIT_PROVIDER	This service adds a new provider to the system.
DELETE_PROVIDER	This service deletes the specified provider.
ENABLE_DISABLE_PROVIDER	This service reverses the provider enable state. If the provider is enabled, the service disables the provider. If the provider is disabled, the service enables the provider.
GET_ADD_EDIT_PROVIDER_FORM	This form return service creates a provider form if one does not exist or updates an existing provider form.
GET_ALL_PROVIDERS	This service returns the provider list containing all providers.
GET_PROVIDER_INFO	This service retrieves provider information.
NOTIFY_CHANGE	This service notifies the provider of changes to the content item.
REQUEST_SECURITYINFO	This service retrieves security information.
TEST_PROVIDER	This service tests the defined provider

AVAILABLE SERVICES

ADD_ALIAS

Description

This service adds alias information.

- ❖ Given an alias name and a defined user, the service adds alias information.
- ❖ The most likely errors are mismatched parameters or when a user or alias already exists in the system. The service throws reasonable exceptions for display to the user in these situations.
- ❖ If you add an alias using the applet, you must add at least one user. However, you do not need to add a user using IdcCommandX. To add users using IdcCommandX, the optional parameter `AliasUsersString` must be included.

Required Parameters

These parameters must be specified:

Parameter	Description
dAlias	The alias name.
dAliasDescription	The alias description.
IdcService	Must be set to <code>ADD_ALIAS</code> .

Optional Parameters

This optional parameter may be specified:

Parameter	Description
AliasUsersString	The alias user or users. To add multiple users, use the \n escape sequence between each user name. For example: AliasUsersString=sysadmin\nuser1

Example

To create an alias and add a user the required parameters will be:

```
IdcService=ADD_ALIAS
```

```
dAlias=my_alias
```

```
dAliasDescription=admin user
```

```
AliasUsersString=sysadmin
```

ADD_ARCHIVE

Description

This service adds an archive to the collection.

- ❖ Given an archive name and a defined instance menu label, the service adds an archive to the collection.
- ❖ The IDC_Name is normally the name of the collection. If you are using the default collection then this is the name of the content server instance.
- ❖ The most likely errors are mismatched parameters or when an archive name is not unique. If this service is unable to execute, this message is displayed to the user: *Unable to add archive "{aArchiveName}" to the collection "{InstanceMenuLabel}"*.



Caution: Using duplicate IDC_Name collection names will cause data corruption. The Archiver cannot be used to move or copy data between two instances that share the same IDC_Name. To do so will corrupt the data on the target system.

Required Parameters

These parameters must be specified:

Parameter	Description
aArchiveDescription	The menu label for the server instance (used on the interface). For example, Master_on_server01.
aArchiveName	The archive name.
IDC_Name	The name of the content server instance.
IdcService	Must be set to ADD_ARCHIVE.

Example

To add an archive to a master content server, the required parameters will be:

```
IdcService=ADD_ARCHIVE
```

```
IDC_Name=Master_on_server01
```

```
aArchiveName=archive_test
```

```
aArchiveDescription=this is an archive test
```

ADD_COLLECTION

Description

This service adds a collection.

- ❖ Given a defined instance menu label, the service adds a collection.
- ❖ The IDC_Name is normally the name of the collection. If you are using the default collection then this is the name of the content server instance.
- ❖ The most likely error is an instance menu label that is not unique. If this service is unable to execute, this message is displayed to the user: *Unable to add collection "{InstanceMenuLabel}"*.
- ❖ If you are adding a new collection, you should add the new name to that collection using the IDC_Name parameter; otherwise the error message "the collection already exists" will be returned.



Caution: Using duplicate IDC_Name collection names will cause data corruption. The Archiver cannot be used to move or copy data between two instances that share the same IDC_Name. To do so will corrupt the data on the target system.

Required Parameters

These parameters must be specified:

Parameter	Description
aCollectionLocation	The absolute path to the collection location. Use the forward slash as the file separator.
aVaultDir	The absolute path to the vault. Use the forward slash as the file separator.

Parameter	Description
aWeblayoutDir	The absolute path to the weblayout directory. Use the forward slash as the file separator.
IDC_Name	The name of the content server instance.
IdcService	Must be set to ADD_COLLECTION.

Example

```
IdcService=ADD_COLLECTION  
IDC_Name=new_collection  
aCollectionLocation=c:/stellent/archives  
aVaultDir=c:/stellent/value  
aWeblayoutDir=c:/stellent/weblayout
```

ADD_DOC_ACCOUNT

Description

This service adds a new account.

- ❖ Given a file account name, the service adds a new account to the system.
- ❖ The most likely error is an account name that is not unique. The service throws reasonable exceptions for display to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
dDocAccount	The security account for the content item.
IdcService	Must be set to ADD_DOC_ACCOUNT.

Example

```
IdcService=ADD_DOC_COLLECTION  
dDocAccount=newaccount
```

ADD_DOCEXTENSION

Description

This service adds a file extension.

- ❖ Given a defined file extension, the service adds the extension to an existing file.
- ❖ The most likely error is when a matching file with that extension already exists in the system. The service throws reasonable exceptions for display to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
dExtension	The file extension such as HCSF, DOC, TXT.
dFormat	The formatting process used to create the web viewable version of the content. For example, application/hcsf or application/doc.
IdcService	Must be set to ADD_DOCEXTENSION.

Example

```
IdcService=ADD_DOCEXTENSION
dExtension=doc
dFormat=application/doc
```

ADD_DOCFORMAT

Description

This service adds a new file format.

- ❖ Given a defined file format name, the service adds a new file format to the system.
- ❖ The most likely error is when the file format already exists in the system. The service throws reasonable exceptions for display to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
dConversion	The conversion algorithm is determined by the parameter dConversion. If the conversion type is not "PASSTHRU" or "NOCONVERSION" (default value) and the content is not web viewable or has a publish state, the content gets added to the queue to get converted. Otherwise, the server looks for the web-viewable (if applicable) and the vault file and updates the processing state to say that the file has already been converted. If the file mime-type is Application/FDF, the server will set the conversion to "EXCHANGE-FDF".
dDescription	The provided description.
dFormat	The formatting process used to create the web viewable version of the content. For example, application/hcsf or application/doc.
IdcService	Must be set to ADD_DOCFORMAT.

Example

```
IdcService=ADD_DOCFORMAT  
dFormat=application/doc  
dDescription=mword  
dConversion=PASSTHRU
```

ADD_DOCTYPE

Description

This service adds a new file type.

- ❖ Given a file format type, the service adds a new file type to the system.
- ❖ The most likely error is when the content item format type name is not unique. The service throws reasonable exceptions for display to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
dDescription	The provided description.
dDocType	The content item type.
dGif	The name of the GIF image with file extension.
IdcService	Must be set to ADD_DOCTYPE.

Example

```
IdcService=ADD_DOCTYPE
dDocType=MYTEST
dDescription=My Description.
dGif=adeng.gif
```

ADD_EDIT_PROVIDER

Description

This service adds a new provider.

- ❖ Given a credit provider name, the service adds a new provider to the system.
- ❖ The most likely error is when the provider name is not unique. If this service is unable to execute, this message is displayed to the user: *Unable to add or edit provider "{pName}"*.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to ADD_EDIT_PROVIDER.
pDescription	The provided description for the provider.
pName	The provider name.
ProviderClass	The provider class.

Parameter	Description
pType	<p>The provider type:</p> <ul style="list-style-type: none"> ❖ Database—An information repository server that provides an API for connecting and communicating with it. This retrieves information and enables information to be changed in the database. Examples of this type are system databases and LDAP. ❖ Incoming—A connection initiated from an outside entity like a browser or client application. The server listens and is aware of incoming connections ❖ Outgoing—A connection initiated to an outside entity. You can use this type to communicate between Content Server instances. ❖ Preview—The API that establishes connections between content server and Preview technology like the DTM server for HTML Preview and Content Categorizer.

Example

```
IdcService=ADD_EDIT_PROVIDER
pName=admin_provider
pType=Database
pDescription=provider description
ProviderClass=providerclass
```

ADD_GROUP

Description

This service adds a group.

- ❖ Given a group name, the service adds a group to the system.
- ❖ The most likely error is when the group name is not unique. The service throws reasonable exceptions for display to the user.



Important: The value `dPrivilege` defines the privilege setting. If this value is set to 0 only admin has RWDA privilege on the group, no other roles will be able to access the group. If the value is set to 15 then all the roles get RWDA privilege on the group. For more information about numbering of privilege, please refer to the *Content Server Security Guide*.

Required Parameters

These parameters must be specified:

Parameter	Description
<code>dDescription</code>	The provided description.
<code>dGroupName</code>	The defined group name.
<code>dPrivilege</code>	The privilege setting.
<code>IdcService</code>	Must be set to <code>ADD_GROUP</code> .

Example

```
IdcService=ADD_GROUP
dGroupName=NEW_GROUP
dPrivilege=15
```

IdcCommand Services

dDescription=admin privileges

ADD_METADEF

Description

This service adds a new meta field.

- ❖ The service adds a new meta definition field to the system.
- ❖ For example, the format for *dOptionListType* may look like this:

```
choice = strict choice
chunval = strict choice unvalidated
combo = edit and choice
multi = multi select
```

- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to add new meta definition field.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to ADD_METADEF.

Example

```
IdcService=ADD_METADEF
```

ADD_PROBLEMREPORT

Description

This workflow related service adds a problem report to the content item.

- ❖ Content Publisher related service used on Staging Workflow.
- ❖ Given a file name and a problem report author, the service adds a problem report to the content item.
- ❖ The most likely errors are mismatched parameters, when the system is unable to add a problem report to the content item, or when the content item no longer exists in the system. The service throws reasonable exceptions for display to the user in these situations.



Note: Refer to the Content Publisher documentation for additional information on Staging Workflow

Required Parameters

These parameters must be specified:

Parameter	Description
dDocName	The content item identifier (Content ID).
dID	The generated content item revision ID.
dPrCaption	The caption for the problem report.
dPrSeverity	The problem report severity level such as CRITICAL, MODERATE, MINOR. This option has no bearing on how a problem report is handled within the workflow process.
dPrState	The problem report state such as OPEN, CLOSED, FIXED.

Parameter	Description
IdcService	Must be set to ADD_PROBLEMREPORT.



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Example

```
IdcService=ADD_PROBLEMREPORT  
dPrCaption=Problem report for this content.  
dDocName=PublicDoc1_ia3c488971  
dID=67  
dPrState=OPEN  
dPrSeverity=MINOR
```

ADD_ROLE

Description

This service adds a role.

- ❖ Given a role name, the service adds a new role.
- ❖ The most likely error is when the role name is not unique. The service throws reasonable exceptions for display to the user.



Important: The value `dPrivilege` defines the privilege setting. If this value is set to 0 only admin has RWDA privilege on the group, no other roles will be able to access the group. If the value is set to 15 then all the roles get RWDA privilege on the group. For more information about numbering of privilege, please refer to the *Content Server Security Guide*.

Required Parameters

These parameters must be specified:

Parameter	Description
<code>dPrivilage</code>	The privilage setting.
<code>dRoleName</code>	The unique role name.
<code>IdcService</code>	Must be set to <code>ADD_ROLE</code> .

Example

```
IdcService=ADD_ROLE
dRoleName=specialuser
dPrivilege=15
```

ADD_USER

Description

This service adds a new user to the system.

- ❖ Given a user name, the service determines if the user is in the system. If the user does not exist, the service will add the user.
- ❖ The most likely error is when the user name is not unique. If this service is unable to execute, an error message is displayed to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
dName	The unique name.
dUserAuthType	The user authorization type. This value must be set to either LOCAL or GLOBAL.
IdcService	Must be set to ADD_USER.

Optional Parameters

These optional parameters may be specified:

Parameter	Description
dEmail	The email address for the user.
dFullName	The full name of the user.
dPassword	The password for the user.

Optional Attribute Information

A result set containing the user's attribute information and referencing the roles the user belongs to and the accounts the user has access to. Attribute information consists of a list of three comma-separated strings. The first string indicates the type of attribute, the second the name of the attribute, and the third is the access number for *accounts* or default entry for *role*.



Important: The user attribute information is not pre-defined. The user by default will belong to no roles or accounts, and will become a guest in the system.

Attribute Information	Description
Access Number	The access number determines the level of access or privileges assigned to the user
Attribute Name	The attribute name is the name of the <i>role</i> or <i>account</i> to be assigned. For example, admin, contributor, or editor may be assigned.
Attribute Type	The attribute types consists of <i>role</i> or <i>account</i>

Access Number

These access numbers can be assigned to the user.

Access Level Flags	Description
1	Read only.
3	Read and write.
7	Read, write, and delete.
15	Administrative privileges.

Attribute Name

A user can belong to multiple roles and accounts, there may be multiple role and account information strings separated by commas in the attribute information column.

- ❖ If the user is to have the admin role, define the user attribute information as follows:

```
@ResultSet UserAttribInfo
2
dUserName
AttributeInfo
jsmith
role,contributor,15
@end
```

- ❖ If the user is to belong to both the contributor and editor roles and have read privilege on the account books, define the user attribute information as:

```
@ResultSet UserAttribInfo
2
dUserName
AttributeInfo
jsmith
role,contributor,15,role,editor,15,account,books,1
@end
```

Attribute Type

When defining a role, the first string specifies that this is a role attribute, the second string is the name of the role, and the third is the default entry of 15.

When defining an account, the first string specifies that this is an account attribute, the second string is the name of the account, and the third is the access level.

- ❖ For an attribute role, the information is in the form:

```
role,contributor,15
```

- ❖ For an attribute account where the access level determines the users rights to the named account, the information is in the form:

```
account,books,1
```

Example

- ❖ IdcCommand command file format:

```
IdcService=ADD_USER  
dName=specialuser  
dUserAuthType=LOCAL
```

- ❖ Serialized HDA format with optional parameters:

```
@Properties LocalData  
IdcService=ADD_USER  
dName=jsmith  
dUserAuthType=LOCAL  
dFullName=Jennifer Smith  
dPassword=password  
dEmail=jsmith@stellent.com  
@end  
@ResultSet UserAttribInfo  
2  
dUserName  
AttributeInfo
```

```
jsmith  
role,admin,15,role,contributor,15  
@end
```

ADD_WF_TEMPLATE

Description

This workflow related service adds a new workflow template.

- ❖ Given a workflow template name, the service adds a new workflow template to the system.
- ❖ The most likely error is when the workflow template name is not unique or when the system is unable to retrieve the workflow template list. The service throws reasonable exceptions for display to the user in these situations.

Required Parameters

These parameters must be specified:

Parameter	Description
dWfTemplateDescription	The workflow template description.
dWfTemplateName	The workflow template name.
IdcService	Must be set to ADD_WF_TEMPLATE.

Example

```
IdcService=ADD_WF_TEMPLATE
dWfTemplateName=mark_temp
dWfTemplateDescription=marketing template
```

ADD_WORKFLOW

Description

This workflow related service adds a new workflow.

- ❖ Given a unique workflow name, the service adds a new workflow to the system.
- ❖ This service is executed by the Workflow Admin applet. Workflows and sub-workflows can be added, edited, enabled, disabled, and deleted from the Workflow Admin administration applet.
- ❖ Workflow Types---From a workflow administrator's point of view, there are three types of workflows:
 - A Basic workflow defines the review process for specific content items, and must be initiated manually.
 - In a Criteria workflow, a file enters the workflow automatically upon checkin when its metadata matches predefined criteria.
 - A sub-workflow is a workflow that does not have an initial contribution step. A file can enter a sub-workflow only through a jump from a Criteria workflow.
- ❖ Each workflow can include multiple review and notification steps, and multiple reviewers can be assigned to approve or reject the file at each step. For each step in a workflow, a set of users and a step type must be defined. The users defined for a step can perform only the tasks allowed for that step type.
- ❖ The most likely error is when the workflow name is not unique. The service throws reasonable exceptions for display to the user.



Note: If the security group of a checked in file does not match the security group of the workflow, the file will not enter the workflow.

Required Parameters

These parameters must be specified:

Parameter	Description
dSecurityGroup	The security group such as PUBLIC or SECURE.
dWfCriteriaName	The workflow criteria name.
dWfCriteriaOperator	The workflow criteria operator such as MATCHES, STARTS, ENDS
dWfCriteriaValue	The workflow criteria value.
dWfDescription	The workflow description.
dWfName	The workflow name.
dWfType	The workflow type such as BASIC or CRITERIA.
IdcService	Must be set to ADD_WORKFLOW.



Note: Refer to the Content Publisher documentation for additional information on workflow types and workflow criteria.

Example

```
IdcService=ADD_WORKFLOW
dWfName=test_workflow
dSecurityGroup=Public
dWfType=Criteria
dWfCriteriaName=dDocAuthor
dWfCriteriaOperator=matches
dWfCriteriaValue=sysadmin
dWfDescription=testing workflow
```

ADD_WORKFLOWALIAS

Description

This workflow related service adds a new workflow alias.

- ❖ Adds contributors in a basic workflow.
- ❖ The most likely errors are mismatched parameters or when the named workflow does not exist. The service throws reasonable exceptions for display to the user in these situations.
- ❖ The dWfStepID parameter is required for this service. The content server automatically assigns dWfStepID for that contribution stage.



Note: Refer to the Content Publisher documentation for additional information.

Required Parameters

These parameters must be specified:

Parameter	Description
aliases	The chosen alias name.
dAliasType	The assigned alias type.
dWfID	The workflow ID.
dWfName	The workflow name.
dWfStepID	The workflow contribution stage ID. This auto-generated value can be retrieved from the database table.
IdcService	Must be set to ADD_WORKFLOWALIAS.

Example

```
IdcService=ADD_WORKFLOWALIAS
```

```
dWfName=C10
```

```
dWfID=12
```

```
dAliasType=user
```

```
aliases=sammy
```

```
dWfStepID=26
```

ADD_WORKFLOWDOCUMENT

Description

This workflow related service adds a new content item to a basic workflow.

- ❖ Given a defined workflow name and a defined file name, the service adds a new content item to the workflow.
- ❖ The most likely errors are mismatched parameters, when the named workflow does not exist, or when the named file does not exist. The service throws reasonable exceptions for display to the user in these situations.
- ❖ This service is called from the Workflow Admin applet by creating a basic workflow and clicking New in the Content section.
- ❖ This service executes the sub-service `ADD_WORKFLOWDOCUMENT_SUB`. This sub-service is a workflow related service that adds a revised content item to the workflow.

Required Parameters

These parameters must be specified:

Parameter	Description
dDocName	The content item identifier (Content ID).
dWfName	The workflow name.
IdcService	Must be set to <code>ADD_WORKFLOWDOCUMENT</code> .



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

IdcCommand Services

Example

```
IdcService=ADD_WORKFLOWDOCUMENT
```

```
dWfName=C10
```

```
dDocName=test1234
```

ADD_WORKFLOWDOCUMENTS

Description

This workflow related service adds all file revisions to the workflow.

- ❖ Given a defined workflow name and a defined file name, the service adds all file revisions to the workflow.
- ❖ This service is called from the Workflow Admin applet by creating a basic workflow and clicking *Select* in the Content section.
- ❖ Workflows and sub-workflows can be added, edited, enabled, disabled, and deleted from the Workflow Admin administration applet.
- ❖ This service executes the sub-service `ADD_WORKFLOWDOCUMENT_SUB`. This sub-service is a workflow related service that adds a revised content item to the workflow.

Required Parameters

These parameters must be specified:

Parameter	Description
<code>dataSource</code>	The data source.
<code>dDocName</code>	The content item identifier (Content ID).
<code>dWfDirectory</code>	The workflow directory setting.
<code>dWfName</code>	The workflow name.
<code>dWfType</code>	The workflow type such as <code>BASIC</code> or <code>CRITERIA</code> .
<code>IdcService</code>	Must be set to <code>ADD_WORKFLOWDOCUMENTS</code> .
<code>resultName</code>	The result set name.
<code>whereClause</code>	The programmatic clause.



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Example

```
IdcService=ADD_WORKFLOWDOCUMENTS  
dWfName=C10  
dDocName=test_000035  
dataSource=Documents  
resultName=DOCUMENTS  
dWfType=Basic  
whereClause=dSubscriptionType='test'  
dWfDirectory=public
```

ADD_WORKFLOWSTEP

Description

This workflow related service adds a new step to the workflow.

- ❖ These parameters can be executed in a browser environment using the Workflow Admin applet. For example, the `dWfStepName`, `dWfStepDescription`, and `dWfStepType` entries correspond to the *Name* entry, the *Description* field, and the *Type* drop-down list of the Workflow Admin applet Edit Step Reviewer screen.
- ❖ Setting `dWfStepWeight` to a numeric value is the same as entering a value in the field associated with the “At least this many reviewers” option for Workflow Steps in the Workflow Admin applet. If defined, the workflow passes to the next step as soon as the number of users specified in `dWfStepWeight` have approved the revision.
- ❖ Setting `dWfStepIsAll` to `TRUE` (1) is the same as enabling the “All Reviewers” option for Workflow Steps in the Workflow Admin applet. If set to `TRUE`, all users assigned to the step must approve the revision before the workflow passes to the next step.
- ❖ The most likely errors are mismatched parameters, when the named workflow is not found in the system, or when the named step is not unique. The service throws reasonable exceptions for display to the user in these situations.

Required Parameters

These parameters must be specified:

Parameter	Description
<code>dWfID</code>	The workflow ID.

Parameter	Description
dWfName	The workflow name.
dWfStepDescription	The provided step description.
dWfStepIsAll	<p>Enables the all reviewer option.</p> <ul style="list-style-type: none"> ❖ Set to TRUE (1) to require all users assigned to the step to approve the revision before the workflow passes to the next step. ❖ Set to FALSE (0) to enable the dWfStepWeight limited reviewer option.
dWfStepName	The workflow step name. Typically this is the type of review (such as <i>initial review</i> or <i>copy edit</i>) or the function of the reviewer (such as <i>manager</i> or <i>copy editor</i>).
dWfStepType	<p>The workflow step type:</p> <ul style="list-style-type: none"> ❖ Reviewer—Approves or rejects the revision. ❖ Reviewer/Contributor—Can edit the revision and approves or rejects it.

Parameter	Description
dWfStepWeight	<p>Defines the number of reviewers for the limited reviewer option.</p> <ul style="list-style-type: none"> ❖ Enter a numeric value for the number of reviewers. The workflow passes to the next step as soon as the number of users specified have approved the revision. ❖ You can enter zero (0) in this field to notify reviewers that the revision has reached the step, but reviewers will not be able to approve, reject, or edit the revision at that step. The workflow will pass to the next step automatically. ❖ The parameter dWfStepIsAll is initially evaluated. If dWfStepIsAll is set to FALSE, this option is enabled.
IdcService	Must be set to ADD_WORKFLOWSTEP.

Optional Parameters

These parameters may be specified:

Parameter	Description
wfEntryScript	The programmatic entry script.
wfExitScript	The programmatic exit script.
wfUpdateScript	The programmatic update script.



Note: Programmatic scripts must be placed within <\$ and \$> delimiters. For example, <\$if ConditionExpression\$>.

Example

```
IdcService=ADD_WORKFLOWSTEP
dWfName=test_workflow
dWfStepName=edit
dWfID=4
dWfStepDescription=this test
dWfStepType=Reviewer/Contributor
dWfStepIsAll=0
dWfStepWeight=1
wfEntryScript=
wfExitScript=
wfUpdateScript=
```

APPLET_DOCINFO

Description

This service retrieves content item information.

- ❖ Given a defined content item name, the service retrieves content item information for the named file.
- ❖ The most likely error is when the content item no longer exists in the system. The service throws reasonable exceptions for display to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
dID	The generated content item revision ID.
IdcService	Must be set to APPLET_DOCINFO

Example

```
IdcService=APPLET_DOCINFO
```

```
dID=47
```

CANCEL_ARCHIVE

Description

This service cancels an archive request.

- ❖ The service cancels an archive request currently running on the system.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to cancel archiving.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to CANCEL_ARCHIVE.

Example

```
IdcService= CANCEL_ARCHIVE
```

CANCEL_SEARCH_INDEX

Description

This service cancels a search indexing session.

- ❖ The service cancels a search indexing session currently running on the system.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to cancel search indexing session.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to CANCEL_SEARCH_INDEX.

Example

```
IdcService=CANCEL_SEARCH_INDEX
```

CHECK_USER_CREDENTIALS

Description

This service checks the credentials of the user.

- ❖ The service checks the credentials of the currently logged on user. For example, this is the format of the command with required parameters when passed in a browser:

```
http://localhost/intradoc-cgi/
idc_cgi_isapi.dll?IdcService=CHECK_USER_CREDENTIALS&getU
se rInfo=1&userName=sysadmin&isJava=1
```

- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to check credentials of user.

Required Parameters

These parameters must be specified:

Parameter	Description
dGetUserInfo	This entry tells the server executing the request if it should retrieve the users extended information (i.e. - email, full name, locale, user type). This entry must be set to FALSE (0) or TRUE (1). Setting this entry to TRUE retrieve the users extended information.
dUserName	The user attribute name. This is the name of the <i>role</i> or <i>account</i> assigned. For example, admin, contributor, or editor.
IdcService	Must be set to CHECK_USER_CREDENTIALS.

Example

```
IdcService=CHECK_USER_CREDENTIALS  
userName=admin  
GetUserInfo=0
```

CHECKIN_ARCHIVE

Description

This service is used to check in a content item into an archive.

- ❖ This service will execute another service based on a number of parameters and conditional variables.
 - DELETE_BYREV - If the Action parameter is "delete" and the content item revision exists in the system.
 - DELETE_BYCLASS - If the Action parameter is "delete" and the content item revision does not exist in the system.
 - CHECKIN_NEW_SUB - If the content item does not exist in the system, the Action parameter is not "delete", and the conditional variable IsPublish is set to TRUE.
 - INSERT_NEW - If the content item does not exist in the system, the Action parameter is not "delete", and the conditional variable IsPublish is set to FALSE or not specified.
 - UPDATE_DOCINFO_BYREV - If the Action parameter is not "delete" and the primaryFile parameter is not empty. Additionally, the conditional variable IsPublish must be TRUE or the content item revision must exist in the system.
 - UPDATE_BYREV - If the Action parameter is not "delete" and the primaryFile parameter is empty. Additionally, the conditional variable IsPublish must be TRUE or the content item revision must exist in the system.



Important: The *IsPublish* variable is set by the server when a workflow or project is unregistered. This value cannot be set as a parameter or as a configuration entry.

- ❖ If the IsNative parameter is true, we are to execute a native command.
 - DELETE_BYREV - If the content item exists in the Revisions table and action parameter is "delete"
 - UPDATE_BYREV - If the content item exists in the Revisions table, the content has not been deleted, and the action parameter is "update".
 - INSERT_NATIVE - If the content item does not exist in the server and the action parameter is not "delete".
- ❖ If a content item already exists in the system and has not been deleted, the server cannot update or insert the entry; an error will be thrown.
- ❖ The content name (dDocName) has to exist and must not contain any spaces or invalid characters (;\|?:@&=+\\"#%<>*~|[]).
- ❖ If doFileCopy is set to TRUE (1), the file will not be deleted from hard drive after successful checkin, otherwise this service will delete the file.
- ❖ The most likely error is when the content item name is not unique or when the service is unable to check in the named file. If this service is unable to execute, this message is displayed to the user: *Content item "{dDocName}" was not successfully checked in*
- ❖ A primary file is required. If you do not want to checkin a primary file, and want to checkin only metadata then an additional parameter must be included and a configuration entry added in the content server.

Required additional parameter (metadata checkin):

```
createPrimaryMetaFile=true
```

Required content server configuration entry (metadata checkin):

```
AllowPrimaryMetaFile=true
```

Example:

```
IdcService=CHECKIN_ARCHIVE
```

```
Action=insert
```

```
dDocTitle=my_test  
dDocAuthor=sysadmin  
dDocType=ADACCT  
dSecurityGroup=Public  
createPrimaryMetaFile=true
```

Required Parameters

These parameters must be specified:

Parameter	Description
Action	Must be set to Insert.
dDocAuthor	The content item author (contributor).
dDocName	The content item identifier (Content ID).
dDocTitle	The content item title.
doFileCopy	Set this flag to TRUE (1) or the file will be removed from your hard drive after checkin.
dSecurityGroup	The security group such as PUBLIC or SECURE.
IdcService	Must be set to CHECKIN_ARCHIVE.

Parameter	Description
primaryFile	<p>The absolute path to the location of the file as seen from the server. Use the forward slash as the file separator.</p> <p>A primary file must be specified unless checking in metadata only. If an alternate file is specified with the primary file, the content refinery will convert the alternate file. Otherwise, the primary file will be converted.</p> <ul style="list-style-type: none"> ❖ If a primary file is not specified, a metafile can be used in its place. Only one metafile can exist though for each content item (i.e. a primary AND alternate meta file cannot co-exist). ❖ If both a primary and alternate file is specified, their extensions must be different.

Additional Parameters

This parameter may be required:

Parameter	Description
dDocAccount	<p>The security account for the content item.</p> <p>If you have accounts enabled, you must pass this parameter.</p>

Optional Parameters

These optional parameters may be specified:

Parameter	Description
alternateFile	<p>The alternate file for conversion.</p> <ul style="list-style-type: none"> ❖ Only one metafile can exist though for each content item (a primary AND alternate meta file cannot co-exist.) ❖ If an alternate file is specified with the primary file, the content refinery will convert the alternate file. Otherwise, the primary file will be converted.
dCreateDate	The date the content item was created. By default, this is the current date.
doDocSecurityCheck	This flag enables document security check. By default this entry is FALSE.
dPublishState	The publish state. If the content item exists in the system dPublishState must be empty.
dReleaseState	The release state (used to indicate the web state of the revision).
dRevLabel	The revision label for the content item. If set, the label will be used to locate the specified revision.

Parameter	Description
IsNative	<p>This is a configuration entry but can be specified as a parameter.</p> <ul style="list-style-type: none"> ❖ If the IsNative parameter is true and the content item exists in the system, the content name (dDocName) provided must coincide with the content name in the database. Otherwise, the command is not native and the service will be determined as if the IsNative parameter is false. ❖ If the IsNative parameter is true and the Action parameter is "delete", then the content item must exist in the system. Otherwise, the command is not native and the service will be determined as if the IsNative parameter is false.
webViewableFile	<p>If the content is marked as web viewable the file format is determined by the parameter webViewableFile:format and the extension is determined by the parameter dWebExtension. Otherwise, the extension and file format are determined by the parameters dExtension and dFormat, respectively. The user can override the file format and extension by overriding these parameters.</p>



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Example

```
IdcService=CHECKIN_ARCHIVE
Action=insert
```

IdcCommand Services

```
dDocAuthor=user1  
dDocName=test  
dDocTitle=new content  
dSecurityGroup=Public  
primaryFile=c:/test.txt  
doFileCopy=true
```

CHECKIN_ARCHIVE_NO_NOTIFY

Description

This service has the same functionality as CHECKIN_ARCHIVE, except that it suppresses content release notifications and checks the security info of the document to ensure the user has sufficient rights.

Required Parameters

Refer to CHECKIN_ARCHIVE for required and optional parameters.

Example

```
IdcService=CHECKIN_ARCHIVE_NO_NOTIFY  
Action=insert  
dDocAuthor=user1  
dDocName=test  
dDocTitle=new content  
dSecurityGroup=Public  
primaryFile=c:/test.txt  
doFileCopy=true
```

CHECKIN_BYNAME

Description

This service is used to check in a content item from the content name or content ID.

- ❖ If the content uses entity security, the user must have at least write privilege granted by the entities, unless the user has administrator rights to the security group being assigned to the document.
- ❖ If the content name does not allow for auto generation, the content name (dDocName) must exist and must not contain any spaces or invalid characters (;/\?:@&=+\\"#%<>*~|[]).
- ❖ If the number of uniquely differentiated content items (items with different content names) is greater than 50, a license feature check is performed to see if the server is licensed to allow unlimited documents. Even if a particular content item is deleted, it does not decrement the count.
- ❖ If the content is in a workflow, the user must have sufficient access to that step.
- ❖ A user must have Administrator privileges to the content item's security settings to change the security group (dSecurityGroup), account (dDocAccount), or author (dDocAuthor).
- ❖ Any Idoc script in the auto prefix is evaluated after the new revision class ID is generated and before metadata validation. The evaluated prefix is used to build the content name.
- ❖ If metafile generation is enabled (by setting the AllowPrimaryMetaFile and/or AllowAlternateMetaFile environment variables) and the metafile generation flags (createPrimaryMetaFile and createAlternateMetaFile) are TRUE, a real file cannot be associated with the current content item. Only one metafile can be associated with a content item - if

createPrimaryMetaFile is TRUE, createAlternateMetaFile must be FALSE and vice-versa.

- ❖ If doFileCopy is set to TRUE (1), the file will not be deleted from hard drive after successful checkin, otherwise this service will delete the file.
- ❖ The most likely error is a revision failing to insert or when the refinery was not successfully initiated. If this service is unable to execute, an error message is displayed to the user.

Required Parameters

Either the content item name or the content item revision ID must be specified:

Parameter	Description
dDocAuthor	The content item author (contributor).
dDocName	The content item identifier (Content ID).
dDocTitle	The content item title.
dDocType	The content item type.
doFileCopy	Set this flag to TRUE (1) or the file will be removed from your hard drive after checkin.
dSecurityGroup	The security group such as PUBLIC or SECURE.
IdcService	Must be set to CHECKIN_BYNAME.

Parameter	Description
primaryFile	<p>The absolute path to the location of the file as seen from the server. Use the forward slash as the file separator.</p> <p>A primary file must be specified unless checking in metadata only. If an alternate file is specified with the primary file, the content refinery will convert the alternate file. Otherwise, the primary file will be converted.</p> <ul style="list-style-type: none"> ❖ If a primary file is not specified, a metafile can be used in its place. Only one metafile can exist though for each content item (i.e. a primary AND alternate meta file cannot co-exist). ❖ If both a primary and alternate file is specified, their extensions must be different.

Additional Parameters

This parameter may be required:

Parameter	Description
dDocAccount	<p>The security account for the content item.</p> <p>If you have accounts enabled, you must pass this parameter.</p>

Optional Parameters

These optional parameters may be specified:

Parameter	Description
alternateFile	<p>The alternate file for conversion.</p> <ul style="list-style-type: none"> ❖ Only one metafile can exist though for each content item (a primary AND alternate meta file cannot co-exist.) ❖ If an alternate file is specified with the primary file, the content refinery will convert the alternate file. Otherwise, the primary file will be converted.
AutoNumberPrefix	<p>This is a configuration entry but can be specified as a parameter. The auto prefix script is evaluated after the revision class ID is generated and before metadata validation.</p>
createAlternateMetaFile	<p>To enable alternate metafile generation, the AllowAlternateMetaFile environment value must be TRUE and the parameter createAlternateMetaFile must be TRUE.</p>
createPrimaryMetaFile	<p>To enable primary metafile generation, the AllowPrimaryMetaFile environment value must be TRUE and the parameter createPrimaryMetaFile must be TRUE.</p>

Parameter	Description
dConversion	<p>The conversion algorithm is determined by the parameter dConversion.</p> <p>If the conversion type is not "PASSTHRU" or "NOCONVERSION" (default value) and the content is not web viewable or has a publish state, the content gets added to the queue to get converted. Otherwise, the server looks for the web-viewable (if applicable) and the vault file and updates the processing state to say that the file has already been converted.</p> <p>If the file mime-type is Application/FDF, the server will set the conversion to "EXCHANGE-FDF".</p>
dCreateDate	<p>The date the content item was created. By default, this is the current date.</p>
dExtension	<p>The file extension such as HCSF, DOC, TXT.</p>
dFormat	<p>The formatting process used to create the web viewable version of the content. For example, application/hcsf or application/doc.</p>
dID	<p>The generated content item revision ID.</p>
dInDate	<p>The content release date. The date the content item is to be released to the web. By default, this is the current date.</p> <p>If the content release date (dInDate) is not specified, the creation date (dCreateDate) is used. This value is auto generated if it is not supplied.</p>
dOutDate	<p>The content expiration date. By default, this is blank and does not specify an expiration date.</p> <p>If the content expiration date (dOutDate) is not entered, the value remains empty. This is a valid state.</p>
dPublishState	<p>The publish state. If the content item exists in the system dPublishState must be empty.</p>

Parameter	Description
dReleaseState	The release state (used to indicate the web state of the revision).
dRevClassID	The revision class ID.
dRevLabel	The revision label for the content item. If set, the label will be used to locate the specified revision.
dWfType	The workflow type such as BASIC or CRITERIA.
IsAutoNumber	This is a configuration entry but can be specified as a parameter. If the configuration variable IsAutoNumber is set to TRUE, the unique content name (dDocName) is generated by concatenating the auto prefix with the revision class ID (dRevClassID, the counter of uniquely differentiated content items), pre-filled with leading zeroes to be six digits long. Auto numbering is not performed, however, if the content name (dDocName) is already specified.
IsEditRev	If set to TRUE, this entry checks whether content revision is in a workflow and enables editing.
IsWorkflowInfo	This is a configuration entry but can be specified as a parameter. If set to TRUE, this entry checks for workflow information and checks allowable actions for the workflow steps.
webViewableFile	If a content item has a web viewable file associated with it, the conversion format is the format of the web viewable file (i.e. the parameter webViewableFile:format) and the extension is the web viewable file's extension (i.e. the parameter dWebExtension). Otherwise, the extension and file format are determined by the parameters dExtension and dFormat, respectively. The user can override the file format and extension by setting these additional parameters.



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Example

```
IdcService=CHECKIN_BYNAME  
dDocName=test1000  
dSecurityGroup=public  
dDocAuthor=sysadmin  
dDocType=ADENG  
dDocTitle=another test  
doFileCopy=true  
primaryFile=c:/test.txt
```

CHECKIN_LIST

Description

This service retrieves the check in list.

- ❖ Given a checked out content item, the service retrieves the check in list from the database.
- ❖ The most likely error is when the check in list cannot be retrieved from the database. If this service is unable to execute, this message is displayed to the user: *Unable to retrieve check in list.*

Required Parameters

The only required parameter is IdcService, which gives all the checked out contents.

Parameter	Description
IdcService	Must be set to CHECKIN_LIST.

Example

```
IdcService=CHECKIN_LIST
```

CHECKIN_NEW

Description

This service checks in a new content item.

- ❖ Given a content item name, the service checks in the specified content item.
- ❖ If doFileCopy is set to TRUE (1), the file will not be deleted from hard drive after successful checkin, otherwise this service will delete the file.
- ❖ If you attempt to set the author (dDocAuthor) to someone other than the currently logged in user, you must have Administrator privileges to the security attributes being assigned to the document.
- ❖ If the content uses entity security, the user must have at least write privilege granted by the entities, unless the user has administrator rights to the security group being assigned to the document.
- ❖ If the content name does not allow for auto generation, the content name (dDocName) must exist and must not contain any spaces or invalid characters (;/\?:@&=+\\"#%<>*~|[]).
- ❖ If the number of uniquely differentiated content items (items with different content names) is greater than 50, a license feature check is performed to see if the server is licensed to allow unlimited documents. Even if a particular content item is deleted, it does not decrement the count.
- ❖ Any Idoc script in the auto prefix is evaluated after the new revision class ID is generated and before metadata validation. The evaluated prefix is used to build the content name.
- ❖ If metafile generation is enabled (by setting the AllowPrimaryMetaFile and/or AllowAlternateMetaFile environment variables) and the metafile generation flags (createPrimaryMetaFile and createAlternateMetaFile) are true, a real file cannot be associated with the current content item. Only one

metafile can be associated with a content item - if createPrimaryMetaFile is true, createAlternateMetaFile must be false and vice-versa.

- ❖ The most likely error is when the content item was not properly defined. If this service is unable to execute, this message is displayed to the user:
Content item "{dDocName}" was not successfully checked in.
- ❖ This service executes the sub-service CHECKIN_NEW_SUB. This sub-service checks in a new content item revision.
- ❖ A primary file is required. If you do not want to checkin a primary file, and want to checkin only metadata then an additional parameter must be included and a configuration entry added in the content server.

Required additional parameter (metadata checkin):

```
createPrimaryMetaFile=true
```

Required content server configuration entry (metadata checkin):

```
AllowPrimaryMetaFile=true
```

Required Parameters

These parameters must be specified:

Parameter	Description
dDocAuthor	The content item author (contributor).
dDocName	The content item identifier (Content ID). This is parameter is required if the Auto Content ID option is not enabled (not required if auto-numbering is enabled).
dDocTitle	The content item title.
dDocType	The content item type.
doFileCopy	Set this flag to TRUE (1) or the file will be removed from your hard drive after checkin.

Parameter	Description
dSecurityGroup	The security group such as PUBLIC or SECURE.
IdcService	Must be set to CHECKIN_NEW.
primaryFile	<p>The absolute path to the location of the file as seen from the server. Use the forward slash as the file separator.</p> <p>A primary file must be specified unless checking in metadata only. If an alternate file is specified with the primary file, the content refinery will convert the alternate file. Otherwise, the primary file will be converted.</p> <ul style="list-style-type: none"> ❖ If a primary file is not specified, a metafile can be used in its place. Only one metafile can exist though for each content item (i.e. a primary AND alternate meta file cannot co-exist). ❖ If both a primary and alternate file is specified, their extensions must be different.

Additional Parameters

This parameter may be required:

Parameter	Description
dDocAccount	<p>The security account for the content item.</p> <p>If you have accounts enabled, you must pass this parameter.</p>

Optional Parameters

These optional parameters may be specified:

Parameter	Description
alternateFile	<p>The alternate file for conversion.</p> <ul style="list-style-type: none"> ❖ Only one metafile can exist though for each content item (a primary AND alternate meta file cannot co-exist.) ❖ If an alternate file is specified with the primary file, the content refinery will convert the alternate file. Otherwise, the primary file will be converted.
AutoNumberPrefix	<p>This is a configuration entry but can be specified as a parameter. The auto prefix script is evaluated after the revision class ID is generated and before metadata validation.</p>
createAlternateMetaFile	<p>To enable alternate metafile generation, the AllowAlternateMetaFile environment value must be TRUE and the parameter createAlternateMetaFile must be TRUE.</p>
createPrimaryMetaFile	<p>To enable primary metafile generation, the AllowPrimaryMetaFile environment value must be TRUE and the parameter createPrimaryMetaFile must be TRUE.</p>

Parameter	Description
dConversion	<p>The conversion algorithm is determined by the parameter dConversion.</p> <p>If the conversion type is not "PASSTHRU" or "NOCONVERSION" (default value) and the content is not web viewable or has a publish state, the content gets added to the queue to get converted. Otherwise, the server looks for the web-viewable (if applicable) and the vault file and updates the processing state to say that the file has already been converted.</p> <p>If the file mime-type is Application/FDF, the server will set the conversion to "EXCHANGE-FDF".</p>
dCreateDate	<p>The date the content item was created. By default, this is the current date.</p>
dFormat	<p>The formatting process used to create the web viewable version of the content. For example, application/hcsf or application/doc</p>
dInDate	<p>The content release date. The date the content item is to be released to the web. By default, this is the current date.</p> <p>If the content release date (dInDate) is not specified, the creation date (dCreateDate) is used. This value is auto generated if it is not supplied.</p>
dOutDate	<p>The content expiration date. By default, this is blank and does not specify an expiration date.</p> <p>If the content expiration date (dOutDate) is not entered, the value remains empty. This is a valid state.</p>
dProjectID	<p>The unique project ID generated by content server.</p>

Parameter	Description
dPublishType	<p>The publish type (used with content created by Content Publisher).</p> <p>These values can be assigned:</p> <p>H - Home N - Navigation S - Query results pages P - Pages G - Gallery Graphics C - Contributor embedded graphics O - All others</p>
dRendition	Used to specify an additional rendition (version) of the content item.
dRevClassID	The revision class ID.
dRevLabel	The revision label for the content item. If set, the label will be used to locate the specified revision.
IsAutoNumber	<p>This is a configuration entry but can be specified as a parameter. If the configuration variable IsAutoNumber is set to TRUE, the unique content name (dDocName) is generated by concatenating the auto prefix with the revision class ID (dRevClassID, the counter of uniquely differentiated content items), pre-filled with leading zeroes to be six digits long. Auto numbering is not performed, however, if the content name (dDocName) is already specified.</p>

Parameter	Description
webViewableFile	If a content item has a web viewable file associated with it, the conversion format is the format of the web viewable file (i.e. the parameter <code>webViewableFile:format</code>) and the extension is the web viewable file's extension (i.e. the parameter <code>dWebExtension</code>). Otherwise, the extension and file format are determined by the parameters <code>dExtension</code> and <code>dFormat</code> , respectively. The user can override the file format and extension by setting these additional parameters.



Note: Do not confuse the Content ID (`dDocName`) with the internal content item revision identifier (`dID`). The *dID* is a generated reference to a specific rendition of a content item.

Example

```
IdcService=CHECKIN_NEW
dDocName=test1111
dDocTitle=test information
dDocAuthor=john
dSecurityGroup=public
primaryFile=c:/test.txt
doFileCopy=1
```

CHECKIN_NEW_FORM

Description

This form return related service checks in a new content item form.

- ❖ Given a content item name, the service checks in the specified content item form.
- ❖ This service brings up the checkin form in a browser environment.
- ❖ The most likely error is when the content item was not properly defined. If this service is unable to execute, this message is displayed to the user:
Content item "{dDocName}" was not successfully checked in.
- ❖ This service executes the sub-service CHECKIN_NEW_SUB. This sub-service checks in a new content item revision.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to CHECKIN_NEW_FORM.

Example

IdcService=CHECKIN_NEW_FORM

CHECKIN_SEL

Description

This service checks in a content item revision.

- ❖ The content item must have been checked out for this service to execute.
- ❖ If the content item is in a workflow, the user must have sufficient access to that step.
- ❖ The content name (dDocName) has to exist and must not contain any spaces or invalid characters (;\|?:@&=+\\"#% <>*~|[]).
- ❖ A user must have Administrator privileges to the content item's security settings to change the security group (dSecurityGroup), account (dDocAccount), or author (dDocAuthor).
- ❖ Any IdocScript in the auto prefix is evaluated after the new revision class ID is generated and before metadata validation. The evaluated prefix is used to build the content name.
- ❖ All prior metadata values are inherited from the previous revision even if it is not specified in the service call.
- ❖ If metafile generation is enabled (by setting the AllowPrimaryMetaFile and/or AllowAlternateMetaFile environment variables) and the metafile generation flags (createPrimaryMetaFile and createAlternateMetaFile) are true, a real file cannot be associated with the current content item. Only one metafile can be associated with a content item - if createPrimaryMetaFile is true, createAlternateMetaFile must be false and vice-versa.
- ❖ The content item cannot be in the published state.
- ❖ The current user must be the author of the content item or have administrator privileges to the security attributes of the content item to insert a new revision.

- ❖ If doFileCopy is set to TRUE (1), the file will not be deleted from hard drive after successful checkin, otherwise this service will delete the file.
- ❖ The most likely error is when the content item is no longer in the system. If this service is unable to execute, this message is displayed to the user:
Content item "{dDocName}" was not successfully checked in.
- ❖ This service executes the sub-service CHECKIN_SEL_SUB. This sub-service checks in a previously checked out content item as a revision.

Required Parameters

These parameters must be specified:

Parameter	Description
dDocAuthor	The content item author (contributor).
dDocName	The content item identifier (Content ID). This is parameter is required if the Auto Content ID option is not enabled (not required if auto-numbering is enabled).
dDocTitle	The content item title.
dDocType	The content item type.
dID	The generated content item revision ID.
doFileCopy	Set this flag to TRUE (1) or the file will be removed from your hard drive after checkin.
dRevLabel	The content item revision label.
dSecurityGroup	The security group such as PUBLIC or SECURE.
IdcService	Must be set to CHECKIN_SEL.

Additional Parameters

This parameter may be required:

Parameter	Description
dDocAccount	The security account for the content item. If you have accounts enabled, you must pass this parameter.

Optional Parameters

These optional parameters may be specified:

Parameter	Description
alternateFile	The alternate file for conversion. ❖ Only one metafile can exist though for each content item (a primary AND alternate meta file cannot co-exist.) ❖ If an alternate file is specified with the primary file, the content refinery will convert the alternate file. Otherwise, the primary file will be converted.
AutoNumberPrefix	This is a configuration entry but can be specified as a parameter. The auto prefix script is evaluated after the revision class ID is generated and before metadata validation.
createAlternateMetaFile	To enable alternate metafile generation, the AllowAlternateMetaFile environment value must be TRUE and the parameter createAlternateMetaFile must be TRUE.

Parameter	Description
createPrimaryMetaFile	To enable primary metafile generation, the AllowPrimaryMetaFile environment value must be TRUE and the parameter createPrimaryMetaFile must be TRUE.
dConversion	<p>The conversion algorithm is determined by the parameter dConversion.</p> <p>If the conversion type is not "PASSTHRU" or "NOCONVERSION" (default value) and the content is not web viewable or has a publish state, the content gets added to the queue to get converted. Otherwise, the server looks for the web-viewable (if applicable) and the vault file and updates the processing state to say that the file has already been converted.</p> <p>If the file mime-type is Application/FDF, the server will set the conversion to "EXCHANGE-FDF".</p>
dCreateDate	The date the content item was created. By default, this is the current date.
dFormat	The formatting process used to create the web viewable version of the content. For example, application/hcsf or application/doc.
dInDate	<p>The content release date. The date the content item is to be released to the web. By default, this is the current date.</p> <p>If the content release date (dInDate) is not specified, the creation date (dCreateDate) is used. This value is auto generated if it is not supplied.</p>
dOutDate	<p>The content expiration date. By default, this is blank and does not specify an expiration date.</p> <p>If the content expiration date (dOutDate) is not entered, the value remains empty. This is a valid state.</p>

Parameter	Description
dProjectID	The unique project ID generated by content server.
dPublishState	The publish state. If the content item exists in the system dPublishState must be empty.
dPublishType	<p>The publish type (used with content created by Content Publisher).</p> <p>These values can be assigned:</p> <p>H - Home N - Navigation S - Query results pages P - Pages G - Gallery Graphics C - Contributor embedded graphics O - All others</p>
dRevClassID	The revision class ID.
IsAutoNumber	<p>This is a configuration entry but can be specified as a parameter. If the configuration variable IsAutoNumber is set to TRUE, the unique content name (dDocName) is generated by concatenating the auto prefix with the revision class ID (dRevClassID, the counter of uniquely differentiated content items), pre-filled with leading zeroes to be six digits long. Auto numbering is not performed, however, if the content name (dDocName) is already specified.</p>
IsWorkflowInfo	<p>This is a configuration entry but can be specified as a parameter. If set to TRUE, this entry checks for workflow information and checks allowable actions for the workflow steps.</p>

Parameter	Description
primaryFile	<p>The absolute path to the location of the file as seen from the server. Use the forward slash as the file separator.</p> <p>A primary file must be specified unless checking in metadata only. If an alternate file is specified with the primary file, the content refinery will convert the alternate file. Otherwise, the primary file will be converted.</p> <ul style="list-style-type: none"> ❖ If a primary file is not specified, a metafile can be used in its place. Only one metafile can exist though for each content item (i.e. a primary AND alternate meta file cannot co-exist). ❖ If both a primary and alternate file is specified, their extensions must be different.
webViewableFile	<p>If a content item has a web viewable file associated with it, the conversion format is the format of the web viewable file (i.e. the parameter <code>webViewableFile:format</code>) and the extension is the web viewable file's extension (i.e. the parameter <code>dWebExtension</code>). Otherwise, the extension and file format are determined by the parameters <code>dExtension</code> and <code>dFormat</code>, respectively. The user can override the file format and extension by setting these additional parameters.</p>



Note: Do not confuse the Content ID (`dDocName`) with the internal content item revision identifier (`dID`). The *dID* is a generated reference to a specific rendition of a content item.

Example

```
IdcService=CHECKIN_SEL
```

IdcCommand Services

```
dDocName=test_000036  
dDocTitle=my test  
dDocAuthor=sysadmin  
dDocType=ADENG  
dSecurityGroup=Public  
dID=49  
dRevLabel=1  
doFileCopy=1
```

CHECKIN_SEL_FORM

Description

This form return related service checks in a content item form revision.

- ❖ Given a content item name, the service checks in a content item form revision.
- ❖ This service is usually used in browser environment to check in "checked out" content. It will bring the content check in form for that specific content with pre-filled metadata fields.
- ❖ The most likely error is when the content item is no longer in the system, information about the content item can not be found, or when the system is unable to check revision properties. If this service is unable to execute, this message is displayed to the user: *Unable to build check in form for "{dDocName}"*.

Required Parameters

These parameters must be specified:

Parameter	Description
dID	The generated content item revision ID.
IdcService	Must be set to CHECKIN_SEL_FORM.

Example

```
IdcService=CHECKIN_SEL_FORM
```

```
dID=55
```

CHECKIN_UNIVERSAL

Description

The service performs a content server controlled check in.

- ❖ Determines if the content item is new or already exists in the system by querying the database using the content ID (dDocName) as the key..
- ❖ If the content item exists in the system, the publish state (dPublishState) must be empty.
- ❖ If a revision label (dRevLabel) is specified, this service will check if the content revision exists in the system; an exception is thrown if it exists.
- ❖ This service will dispatch this request to:
 - CHECKIN_NEW SUB - If the content item does not exist in the server.
 - CHECKIN_SEL_SUB - If the content item exists on the system and no valid revision was specified and the content item is checked out.
 - WORKFLOW_CHECKIN_SUB - If the content item exists and is part of a workflow.
- ❖ The most likely errors are mismatched parameters or when the content item was not successfully checked in. If this service is unable to execute, this message is displayed to the user: *Content item "{dDocName}" was not successfully checked in.*

The CHECKIN_UNIVERSAL service is a content server controlled check in. The check in will fall into either a new, selected, or workflow check in process and follow the same logic as a check in through the browser or Repository Manager application. If the content item to be checked in already exists in the system, the content item must be checked out for the check in to succeed.

These are essentially the same sub-services used during a content server controlled check in. However, these sub-services are not called during a

BatchLoad or Archive import. This service will check security to determine if the user has sufficient privilege to perform a check in on the content item and if the content item (if it exists) has been checked out. Also, it will determine if the content item matches a workflow criteria or belongs to an active basic workflow.

If the content item is not found the content item is checked in using the CHECKIN_NEW_SUB sub-service. This sub-service validates the check in data and determines if this content item belongs to a criteria workflow. If the content item already exists in the system and the content item does not belong to a workflow, the CHECKIN_SEL_SUB is used. Otherwise the content item exists and belongs to a workflow and the WORKFLOW_CHECKIN_SUB is used.



Note: All paths use the forward slash (“/”) as the file separator. This is because the backslash (“\”) is an escape character. For example, `primaryFile=d:/temp/myfile.txt` should point to the primary file to check in.

Required Parameters

These parameters must be specified:

Parameter	Description
dDocAuthor	The content item author (contributor).
dDocName	<p>The content item identifier (Content ID).</p> <ul style="list-style-type: none"> ❖ This field is optional if the system has been configured with <code>IsAutoNumber</code> set to <code>TRUE</code>. In this scenario, if the <code>dDocName</code> is not specified, the check in will always be new, and the system will generate a new name for the content item. ❖ Otherwise, if <code>dDocName</code> is specified, the service will use this key to do a look up to determine what type of check in to perform.

Parameter	Description
dDocTitle	The content item title.
dDocType	The content item type.
doFileCopy	Set this flag to TRUE (1) or the file will be removed from your hard drive.
dSecurityGroup	The security group such as PUBLIC or SECURE.
IdcService	Must be set to CHECKIN_UNIVERSAL.
primaryFile	<p>The absolute path to the location of the file as seen from the server. Use the forward slash as the file separator.</p> <p>A primary file must be specified unless checking in metadata only. If an alternate file is specified with the primary file, the content refinery will convert the alternate file. Otherwise, the primary file will be converted.</p> <ul style="list-style-type: none"> ❖ If a primary file is not specified, a metafile can be used in its place. Only one metafile can exist though for each content item (i.e. a primary AND alternate meta file cannot co-exist). ❖ If both a primary and alternate file is specified, their extensions must be different.



Important: Custom metadata fields that are defined must also be specified.

Additional Parameters

This parameter may be required:

Parameter	Description
dDocAccount	The security account for the content item. If you have accounts enabled, you must pass this parameter.

Optional Parameters

These optional parameters may be specified:

Parameter	Description
alternateFile	The alternate file for conversion. <ul style="list-style-type: none"> ❖ Only one metafile can exist though for each content item (a primary AND alternate meta file cannot co-exist.) ❖ If an alternate file is specified with the primary file, the content refinery will convert the alternate file. Otherwise, the primary file will be converted.
dCreateDate	The date the content item was created. By default, this is the current date.
dInDate	The content release date. The date the content item is to be released to the web. By default, this is the current date. If the content release date (dInDate) is not specified, the creation date (dCreateDate) is used. This value is auto generated if it is not supplied.

Parameter	Description
dOutDate	The content expiration date. By default, this is blank and does not specify an expiration date. If the content expiration date (dOutDate) is not entered, the value remains empty. This is a valid state.
dRevLabel	The revision label for the content item. If set, the label will be used to locate the specified revision.
isFinished	Set to TRUE (1) if this is a workflow check-in and you have finished editing it. See WORKFLOW_CHECKIN for additional information.



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Example

❖ IdcCommand command file format:

```
IdcService=CHECKIN_UNIVERSAL
dDocName=adsales
dDocType=ADACCT
dDocTitle=Advertising Sales
dSecurityGroup=Secure
dDocAuthor=user1
dRevLabel=1
dDocType=ADACCT
primaryFile=c:/temp/docs/mydoc.txt
doFileCopy=1
```

```
# If this is a workflow check-in and you have finished
editing it, mark it as finished.
```

```
#isFinished=true
```

```
#Required depending on configuration
```

```
dDocAccount=mainaccount
```

```
#Optional fields:
```

```
#dCreateDate=
```

```
#dInDate=
```

```
#dOutDate=
```

```
#alternateFile=
```

```
#Custom metadata fields:
```

```
xComments=
```

```
xLocation=
```

```
xProjects=
```

❖ Serialized HDA format (check in the content item myDocName):

```
@Properties LocalData
```

```
IdcService=CHECKIN_UNIVERSAL
```

```
doFileCopy=1
```

```
dDocName=myDocName
```

```
dDocTitle=My document title
```

```
dDocType=ADACCT
```

```
dSecurityGroup=Secure
```

```
dDocAuthor=user1
```

IdcCommand Services

```
primaryFile=c:/temp/docs/mydoc.txt  
dDocAccount=mainaccount  
xComments=  
xLocation=  
xProjects=  
@end
```

CHECKOUT

Description

This service check outs the latest revision of the specified content item.

- ❖ Given a content item revision ID, the service attempts to locate the content item in the system.
- ❖ The service will fail if the content item does not exist in the system, if the content item is already checked out, or if the user does not have sufficient privilege to checkout the content item.
- ❖ This service is used from a browser and returns a response page (compare CHECKOUT_BY_NAME).
- ❖ The most likely error is when the content no longer exists in the system or when the system is unable to retrieve revision information. If this service is unable to execute, an error message is displayed to the user.
- ❖ This service executes the sub-service CHECKOUT_SUB. This sub-service checks out a content item revision.



Note: This service only marks the content item as locked. It does not perform a download.

Required Parameters

These parameters must be specified:

Parameter	Description
dID	The generated content item revision ID.
IdcService	Must be set to CHECKOUT.

Optional Parameters

This optional parameter may be specified:

Parameter	Description
dDocTitle	The content item title.

Example

- ❖ IdcCommand command file format:

```
IdcService=CHECKOUT  
dID=55
```

- ❖ Serialized HDA format with optional parameter:

```
@Properties LocalData  
IdcService=CHECKOUT  
dID=55  
dDocTitle=Sample Title  
@end
```

CHECKOUT_BY_NAME

Description

This service checks out the latest revision of the specified content item.

- ❖ Given a content item name, the service attempts to locate the content item in the system.
- ❖ The service will fail if the content item does not exist in the system, if the content item is already checked out, or if the user does not have sufficient privilege to checkout the content item.
- ❖ This service is used by an applet or application. (compare CHECKOUT).
- ❖ The most likely error is a content item name not in the system. If this service is unable to execute, an error message is displayed to the user.
- ❖ This service executes the sub-service CHECKOUT_SUB. This sub-service checks out a content item revision.



Note: This service only marks the content item as locked. It does not perform a download.

Required Parameters

These parameters must be specified:

Parameter	Description
dDocName	The content item identifier (Content ID).
IdcService	Must be set to CHECKOUT_BY_NAME.



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Optional Parameters

This optional parameter may be specified:

Parameter	Description
dDocTitle	The content item title.

Example

- ❖ IdcCommand command file format:

```
IdcService=CHECKOUT_BY_NAME  
dDocName=myDocument
```

- ❖ Serialized HDA format with optional parameter:

```
@Properties LocalData  
IdcService=CHECKOUT_BY_NAME  
dDocName=myDocument  
dDocTitle=Just a title  
@end
```

CHECKOUT_OK

Description

This service checks out a revision

- ❖ Form return service. Given a content item name, the service checks out the current revision.
- ❖ Normally, this service is called in the browser environment when a user checks out content. This displays the check out confirmation page allowing the user to re-checkin the content.
- ❖ The most likely error is when the revision no longer exists or when the system is unable to retrieve revision information. The service throws reasonable exceptions for display to the user in these situations.
- ❖ This service executes the sub service CHECKOUT_SUB. This sub-service checks out a content item revision.

Required Parameters

These parameters must be specified:

Parameter	Description
dID	The generated content item revision ID.
IdcService	Must be set to CHECKOUT_OK.

Example

```
IdcService=CHECKOUT_OK
```

```
dID=48
```

CONFIG_INFO

Description

This Admin Server related service retrieves configuration information.

- ❖ The service retrieves configuration information for the currently active component.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve configuration information.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to CONFIG_INFO.

Example

IdcService=CONFIG_INFO

CONTROL_SEARCH_INDEX

Description

This service is used to update or rebuild the search index.

- ❖ Updates or rebuilds are performed automatically in a background thread.
- ❖ Automatic Update Cycle—Incrementally updates the index database. This is usually not necessary because the Index is automatically updated approximately every five minutes by the server.
- ❖ Collection Rebuild Cycle—The search index is entirely rebuilt, and the old index collection is replaced with a new index collection once the rebuild has successfully completed.



Important: Rebuilding the Search Index is necessary only when you change or add Information Fields. Depending on the quantity and size of your files, this process may take a couple of days. If rebuilding is necessary, rebuild at times of non-peak system usage. Rebuilding is system-intensive.

- ❖ This service is called when changes are made to the Automatic Update Cycle and Collection Rebuild Cycle options provided on the Repository Manager: Indexer Tab.
- ❖ To configure the automatic update cycle or collection rebuild cycle using the Repository Manager, click **Configure** in the Automatic Update Cycle area or Collection Rebuild Cycle area of the Indexer tab. This enables you to configure how the search index automatically indexes new files and revision or how a rebuild of the entire collection indexes new files and revisions.
- ❖ The options defined on the Configure Automatic Update Cycle or Configure Collection Rebuild Cycle screen map to these parameters:
 - Content Items Per Indexer Batch (MaxCollectionSize)

- Content Items Per Checkpoint (IndexerCheckpointCount)
 - Indexer Debug Level (SearchDebugLevel)
 - Indexer Auto Updates (sEnableAutoUpdate)
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to execute indexing control.



Tech Tip: This service can be executed with *getStatus* set to TRUE, to return the defined settings and status information (see example below).

Required Parameters

These parameters must be specified:

Parameter	Description
action	<p>The action to perform. This entry may be set to these values:</p> <ul style="list-style-type: none"> ❖ start—Begins the update/rebuild cycle. ❖ cancel—Stops the update/rebuild cycle, and does not permit a future restart. ❖ suspend—Stops the update/rebuild cycle, and permits a future restart. ❖ restart—Restarts the update/rebuild cycle after a <i>suspend</i> was executed. ❖ setConfiguration—Used when enabling or disabling the update cycle. Setting this value enables you to adjust the batch, checkpoint, debug level, and auto update option.

Parameter	Description
cycleID	<p>The cycle action must be specified as either UPDATE or REBUILD.</p> <ul style="list-style-type: none">❖ The UPDATE cycle is started automatically to bring the search collection up to date with the Content Server repository.❖ The REBUILD cycle creates a new search collection containing all released documents.
IdcService	Must be set to CONTROL_SEARCH_INDEX.

Additional Parameters

These parameters may be required depending on the action to execute (see examples below):

Parameter	Description
GetCurrentIndexingStatus	<p>Setting this parameter to TRUE (1), returns the status of the cycle state:</p> <ul style="list-style-type: none"> ❖ Idle—An indexer cycle is complete. ❖ Active—An indexer cycle is currently running. ❖ Interrupt—An unexpected event, that abruptly ends the indexing cycle. For example, a power, database, or file system failure. ❖ Suspend—The indexing cycle was stopped in a controlled manner. For example using the Suspend button on the Repository Manager: Indexer Tab. ❖ Restart—An interrupted or suspended cycle was started again. If you are restarting after an interrupted cycle, ensure that you have corrected the problem that caused the interrupt to occur. ❖ Cancel—The indexing cycle was stopped with no intent to restart.
getStatus	<p>Setting this parameter to TRUE (1), returns the defined settings and status information.</p>

Parameter	Description
IndexerCheckpointCount	<p>The number of files that will go through all relevant indexing states at a time.</p> <ul style="list-style-type: none"> ❖ You can have multiple batches of files indexed per checkpoint. ❖ To update this entry, the action must be set to <i>setConfiguration</i>.
MaxCollectionSize	<p>The maximum number of files that the search index will process at one time.</p> <ul style="list-style-type: none"> ❖ The default is 25. This means 25 files are indexed together, then the next 25 files are indexed. ❖ You can change this setting to one (1) if you are experiencing problems with the search engine indexing large and complicated files. However, slow system performance will result. ❖ To update this entry, the action must be set to <i>setConfiguration</i>.
PerformProcessConversion	<p>Enables the process conversion when starting or restarting the Indexer rebuild process.</p> <ul style="list-style-type: none"> ❖ Set to TRUE (1) or FALSE (0) ❖ When set to TRUE, the process conversion is performed.

Parameter	Description
SearchDebugLevel	<p>The indexer debug level.</p> <ul style="list-style-type: none"> ❖ The more debug information listed in the server window, the slower the indexing progresses. ❖ To update this entry, the action must be set to <i>setConfiguration</i>. <p>These are the debug levels from the least to the most debug information:</p> <ul style="list-style-type: none"> ❖ none—No information for each file accessed is displayed. ❖ verbose—Displays information for each file accessed. Indicates indexed, ignored, or failed. ❖ debug—Displays the medium level of information. ❖ trace—Displays the lowest level of information. ❖ all—Displays the highest level of information.
sEnableAutoUpdate	<p>Sets the indexer auto update option when enabling or disabling the cycle update.</p> <ul style="list-style-type: none"> ❖ Set to TRUE (1) or FALSE (0). ❖ When set to TRUE, the automatic update is enabled. ❖ To update this entry, the action must be set to <i>setConfiguration</i>.

Example

These examples are for controlling the indexer from another process. This is useful because the indexer should not be running when the Verity database is being optimized and in some configurations it is important to optimize the Verity collection on a regular scheduled basis.

- ❖ Serialized HDA format to return the defined settings and status information:

```
@Properties LocalData
IdcService=CONTROL_SEARCH_INDEX
getStatus=1
@end
```

- ❖ Serialized HDA format to enable the update cycle:

```
@Properties LocalData
IdcService=CONTROL_SEARCH_INDEX
MaxCollectionSize=500
sEnableAutoUpdate=1
cycleID=update
action=setConfiguration
SearchDebugLevel=none
IndexerCheckpointCount=5000
GetCurrentIndexingStatus=1
@end
```

- ❖ Serialized HDA format to disable the update cycle:

```
@Properties LocalData
IdcService=CONTROL_SEARCH_INDEX
MaxCollectionSize=500
sEnableAutoUpdate=0
cycleID=update
```

IdcCommand Services

```
action=setConfiguration
SearchDebugLevel=none
IndexerCheckpointCount=5000
GetCurrentIndexingStatus=1
@end
```

❖ **Serialized HDA format to start the Indexer rebuild:**

```
@Properties LocalData
IdcService=CONTROL_SEARCH_INDEX
cycleID=rebuild
action=start
getStatus=1
GetCurrentIndexingStatus=1
PerformProcessConversion=1
@end
```

❖ **Serialized HDA format to cancel the Indexer rebuild:**

```
@Properties LocalData
IdcService=CONTROL_SEARCH_INDEX
cycleID=rebuild
action=cancel
getStatus=1
GetCurrentIndexingStatus=1
@end
```

❖ **Serialized HDA format to suspend the Indexer rebuild:**

```
@Properties LocalData
IdcService=CONTROL_SEARCH_INDEX
cycleID=rebuild
action=suspend
```

```
getStatus=1  
GetCurrentIndexingStatus=1  
@end
```

- ❖ **Serialized HDA format to restart the Indexer rebuild after being interrupted or suspended:**

```
@Properties LocalData  
IdcService=CONTROL_SEARCH_INDEX  
cycleID=rebuild  
action=restart  
getStatus=1  
GetCurrentIndexingStatus=1  
PerformProcessConversion=1  
@end
```

COPY_ARCHIVE

Description

This service copies an archive into a collection.

- ❖ Given an instance menu label, the service copies the archive into the collection.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to copy archive into the collection "{InstanceMenuLabel}".

Required Parameters

These parameters must be specified:

Parameter	Description
InstanceMenuLabel	The menu label for the server instance (used on the interface). For example, Master_on_server01.
IdcService	Must be set to COPY_ARCHIVE.

Example

```
IdcService=COPY_ARCHIVE
```

```
InstanceMenuLabel=Master_on_server01
```

CREATE_SUBSCRIPTION_TYPE

Description

This service creates a subscription type.

- ❖ Given a subscription type name, the service creates a new subscription type entry.
- ❖ The most likely error is when the subscription type name is not unique. If this service is unable to execute, this message is displayed to the user:
Unable to create subscription type "{scpType}".

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to CREATE_SUBSCRIPTION_TYPE.
scpDescription	The subscription description.
scpEnabled	When set to TRUE (1), enables subscriptions based on criteria.
scpFields	The subscription field type (metadata field). For example, dDocAuthor, dDocType.
scpType	The unique subscription type (provided name).

Example

```
IdcService=CREATE_SUBSCRIPTION_TYPE
scpType=subscription_test
scpFields=dDocType
scpDescription=testing the subscription
```

IdcCommand Services

scpEnabled=1

CRITERIAWORKFLOW_DISABLE

Description

This workflow related service disables a criteria workflow.

- ❖ Given a named workflow, the service disables the criteria workflow.
- ❖ This service is executed by the Workflow Admin applet. Criteria workflows and sub-workflows can be added, edited, enabled, disabled, and deleted from the *Criteria* tab of the Workflow Admin administration applet.
- ❖ The most likely error is when the named workflow does not exist. The service throws reasonable exceptions for display to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
dWfName	The workflow name.
IdcService	Must be set to CRITERIAWORKFLOW_DISABLE.

Example

```
IdcService=CRITERIAWORKFLOW_DISABLE
```

```
dWfName=C10
```

CRITERIAWORKFLOW_ENABLE

Description

This workflow related service enables a criteria workflow.

- ❖ Given a workflow name, the service enables the defined workflow.
- ❖ This service is executed by the Workflow Admin applet. Criteria workflows and sub-workflows can be added, edited, enabled, disabled, and deleted from the *Criteria* tab of the Workflow Admin administration applet.
- ❖ The most likely error is when the named workflow does not exist. The service throws reasonable exceptions for display to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
dWfName	The workflow name.
IdcService	Must be set to CRITERIAWORKFLOW_ENABLE.

Example

```
IdcService=CRITERIAWORKFLOW_ENABLE
dWfName=C10
```

DEL_METADEF

Description

This service deletes the meta data information from the meta definition field.

- ❖ To delete metadata, you must add the "x" prefix to the metadata name.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to delete meta definition field.

Required Parameters

These parameters must be specified:

Parameter	Description
dName	The unique name.
IdcService	Must be set to DEL_METADEF.

Example

```
IdcService=DEL_METADEF
```

```
dName=xTEXT
```

DELETE_ALIAS

Description

This service deletes the alias information.

- ❖ Given an alias name the service deletes alias information.
- ❖ The most likely errors are mismatched parameters, when the alias is being used in the workflow, or when the system is unable to delete the alias or the user. The service throws reasonable exceptions for display to the user in these situations.

Required Parameters

These parameters must be specified:

Parameter	Description
dAlias	The alias name.
IdcService	Must be set to DELETE_ALIAS.

Example

```
IdcService=DELETE_ALIAS  
dAlias=admin_alias
```

DELETE_ARCHIVE

Description

This service deletes an archive from the collection.

- ❖ Given an archive name and the content server instance, the service deletes the specified archive from the collection.
- ❖ Normally, IDC_Name is the name of collection. If you are deleting from the default collection then this name is usually the name of the content server instance.
- ❖ The most likely error is when the specified archive does not exist on the system. If this service is unable to execute, this message is displayed to the user: *Unable to delete archive from collection "{InstanceMenuLabel}"*.



Caution: Using duplicate IDC_Name collection names will cause data corruption. The Archiver cannot be used to move or copy data between two instances that share the same IDC_Name. To do so will corrupt the data on the target system.

Required Parameters

These parameters must be specified:

Parameter	Description
aArchiveName	The archive name.
IDC_Name	The name of the content server instance.
IdcService	Must be set to DELETE_ARCHIVE.

Example

```
IdcService=DELETE_ARCHIVE
```

IdcCommand Services

```
IDC_Name=Master_on_server01  
aArchiveName=archive_test
```

DELETE_BATCH_FILE

Description

This service deletes a batch file from within an archive.

- ❖ Given an archive name, the HDA batch file, and the content server instance, the service deletes the specified archive from the collection.
- ❖ The IDC_Name is normally the name of the collection. If you are deleting from the default collection then this name is usually the name of the content server instance.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to delete batch file.



Caution: Using duplicate IDC_Name collection names will cause data corruption. The Archiver cannot be used to move or copy data between two instances that share the same IDC_Name. To do so will corrupt the data on the target system.

Required Parameters

These parameters must be specified:

Parameter	Description
aArchiveName	The archive name.
aBatchFile	The HDA batch file.
IDC_Name	The name of the content server instance.
IdcService	Must be set to DELETE_BATCH_FILE.

Example

```
IdcService=DELETE_BATCH_FILE
```

IdcCommand Services

IDC_Name=Master_on_server01

aArchiveName=archive_test

aBatchFile=02-jan-16_12.02.06_184/0216120206~1.hda

DELETE_BATCH_FILE_DOCUMENTS

Description

This service deletes all content items from the batch file.

- ❖ Given an archive name, the HDA batch file, and the content server instance, the service deletes all content items from the batch file.
- ❖ The IDC_Name is normally the name of the collection. If you are deleting from the default collection then this name is usually the name of the content server instance.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to delete batch file content.



Caution: Using duplicate IDC_Name collection names will cause data corruption. The Archiver cannot be used to move or copy data between two instances that share the same IDC_Name. To do so will corrupt the data on the target system.

Required Parameters

These parameters must be specified:

Parameter	Description
aArchiveName	The archive name.
aBatchFile	The HDA batch file.
IDC_Name	The name of the content server instance.
IdcService	Must be set to DELETE_BATCH_FILE_DOCUMENTS.

Example

```
IdcService=DELETE_BATCH_FILE_DOCUMENTS
```

```
IDC_Name=Master_on_server01
```

```
aArchiveName=archive_test
```

```
aBatchFile=02-jan-16_16.53.02_289/0216165302~1.hda
```

DELETE_DOC

Description

This service deletes a content item.

- ❖ Given a content item revision ID, the specified content item is deleted from the system.
- ❖ The most likely error is when the content item no longer exists in the system. If this service is unable to execute, an error message is displayed to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
dID	The generated content item revision ID.
IdcService	Must be set to DELETE_DOC.

Example

```
IdcService=DELETE_DOC
```

```
dID=48
```

DELETE_DOC_ACCOUNT

Description

This service deletes a content item account.

- ❖ Given a named content item account name, the service deletes the account.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to delete account "{dDocAccount}".

Required Parameters

These parameters must be specified:

Parameter	Description
dDocAccount	The security account for the content item.
IdcService	Must be set to DELETE_DOC_ACCOUNT

Example

```
IdcService=DELETE_DOC  
dDocAccount=mainaccount
```

DELETE_DOCEXTENSION

Description

This service deletes a file extension.

- ❖ Used to delete file extensions such as .txt, .doc, .gif.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to delete file extension.

Required Parameters

These parameters must be specified:

Parameter	Description
dExtension	The file extension such as HCSF, DOC, TXT.
IdcService	Must be set to DELETE_DOCEXTENSION.

Example

```
IdcService=DELETE_DOCFORMAT  
dExtension=hcsf
```

DELETE_DOCFORMAT

Description

This service deletes the formatting process of the current content item.

- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to delete file format.

Required Parameters

These parameters must be specified:

Parameter	Description
dFormat	The formatting process used to create the web viewable version of the content. For example, application/hcsf or application/doc.
IdcService	Must be set to DELETE_DOCFORMAT

Example

```
IdcService=DELETE_DOCFORMAT
```

```
dFormat=application/hcsf
```

DELETE_DOCTYPE

Description

This service deletes content item types.

- ❖ Given a content item type, the service deletes the defined type.
- ❖ The most likely errors are a named file type that does not exist or when a file of that type still exists in the system. If this service is unable to execute, this message is displayed to the user: *Unable to delete file format "{dDocType}"*

Required Parameters

These parameters must be specified:

Parameter	Description
dDocType	The content item type.
IdcService	Must be set to DELETE_DOCTYPE

Example

```
IdcService=DELETE_DOCTYPE
```

```
dDocType=TEST
```

DELETE_GROUP

Description

This service deletes a group.

- ❖ Given a group name, the service deletes a defined group from the system.
- ❖ The most likely errors are when content items or workflows associated with this group still exist in the system. The service throws reasonable exceptions for display to the user in these situations.

Required Parameters

These parameters must be specified:

Parameter	Description
dGroupName	The defined group name.
IdcService	Must be set to DELETE_GROUP.

Example

IdcService=DELETE_GROUP

dGroupName=admingroup

DELETE_PROBLEMREPORT

Description

This workflow related service deletes a problem report.

- ❖ Given a problem report ID, the service deletes the problem report.
- ❖ The most likely errors are mismatched parameters or when the content item no longer exists in the system. The service throws reasonable exceptions for display to the user in these situations.

Required Parameters

These parameters must be specified:

Parameter	Description
dPrID	The problem report ID
IdcService	Must be set to DELETE_PROBLEMREPORT.

Example

```
IdcService=DELETE_PROBLEMREPORT
```

```
dPrID=1
```

DELETE_PROVIDER

Description

This service deletes a provider.

- ❖ Given a provider name, the service deletes the specified provider.
- ❖ The most likely error is a provider name not in the system. If this service is unable to execute, this message is displayed to the user: *Unable to delete provider "{pName}"*.

Required Parameters

These parameters must be specified:

Parameter	Description
pName	The provider name.
IdcService	Must be set to DELETE_PROVIDER

Example

```
IdcService=DELETE_PROVIDE
```

```
pName=Proxied_2_on_test13
```

DELETE_RESULT_TEMPLATE

Description

This service deletes a result template.

- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to delete result template "{name}".

Required Parameters

These parameters must be specified:

Parameter	Description
name	The template name.
IdcService	Must be set to DELETE_RESULT_TEMPLATE.

Example

```
IdcService=DELETE_RESULT_TEMPLATE  
name=test_template
```

DELETE_REV

Description

This service deletes a revision.

- ❖ Given a content item revision ID, the service deletes the specified revision.
- ❖ This service is used from a browser and returns a response page (compare DELETE_REV_EX).
- ❖ The most likely errors are mismatched parameters, when the content item no longer exists in the system, or when the content item is part of a workflow. If this service is unable to execute, an error message is displayed to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
dID	The generated content item revision ID.
IdcService	Must be set to DELETE_REV.

Example

```
IdcService=DELETE_REV
```

```
dID=51
```

DELETE_REV_EX

Description

This service deletes a revision.

- ❖ Given a content item revision ID, the service deletes the specified revision.
- ❖ This service is used by an applet or application (compare DELETE_REV).
- ❖ The most likely errors are mismatched parameters, when the content item no longer exists in the system, or when the content item is part of a workflow. If this service is unable to execute, an error message is displayed to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
dID	The generated content item revision ID.
IdcService	Must be set to DELETE_REV_EX.

Example

```
IdcService=DELETE_REV_EX  
dID=56
```

DELETE_ROLE

Description

This service deletes a role.

- ❖ Given a role name, the service deletes the specified role.
- ❖ The most likely errors are when the named role does not exist in the system or when a user still has this role assigned. The service throws reasonable exceptions for display to the user in these situations

Required Parameters

These parameters must be specified:

Parameter	Description
dRoleName	The unique role name.
IdcService	Must be set to DELETE_ROLE.

Example

```
IdcService=DELETE_ROLE
```

```
dRoleName=test_role
```

DELETE_SUBSCRIPTION_TYPE

Description

This service deletes a subscription type.

- ❖ Given a named subscription type, the service removes the subscription type.
- ❖ The most likely error is a subscription type that is not in the system. If this service is unable to execute, this message is displayed to the user: *Unable to delete subscription type "{scpType}"*.

Required Parameters

These parameters must be specified:

Parameter	Description
scpType	The unique subscription type (provided name).
IdcService	Must be set to DELETE_SUBSCRIPTION_TYPE.

Example

```
IdcService=DELETE_SUBSCRIPTION_TYPE
```

```
scpType=subscription_test
```

DELETE_USER

Description

This service deletes an existing user.

- ❖ Given a user name, the service deletes the user from the system.
- ❖ The most likely error is when the user has been assigned to an alias. If this service is unable to execute, an error message is displayed to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
dName	The unique name.
IdcService	Must be set to DELETE_USER.

Example

- ❖ IdcCommand command file format:

```
IdcService=DELETE_USER
dName=jsmith
```

- ❖ Serialized HDA format:

```
@Properties LocalData
IdcService=DELETE_USER
dName=jsmith
@end
```

DELETE_WF_TEMPLATE

Description

This workflow related service deletes a workflow template.

- ❖ Given a workflow template name, the service deletes the specified workflow template.
- ❖ The most likely error is when the system is unable to retrieve workflow templates. If this service is unable to execute, this message is displayed to the user: *Unable to delete template "{dWfTemplateName}"*.

Required Parameters

These parameters must be specified:

Parameter	Description
dWfTemplateName	The workflow template name.
IdcService	Must be set to DELETE_WF_TEMPLATE.

Example

```
IdcService=DELETE_WF_TEMPLATE  
dWfTemplateName=template_10
```

DELETE_WFCONTRIBUTORS

Description

This workflow related service deletes workflow contributors.

- ❖ Given a user alias, a workflow step ID, and a workflow name, the service deletes the specified contributor from basic workflow.
- ❖ The dWfStepID parameter is required for this service. The content server automatically assigns dWfStepID for that contribution stage.
- ❖ The most likely errors are mismatched parameters or when the named workflow or alias does not exist. The service throws reasonable exceptions for display to the user in these situations.

Required Parameters

These parameters must be specified:

Parameter	Description
alias	The user alias.
dWfName	The workflow name.
dWfStepID	The workflow contribution stage ID. This auto-generated value can be retrieved from the database table.
IdcService	Must be set to DELETE_WFCONTRIBUTORS.

Example

```
IdcService=DELETE_WFCONTRIBUTORS
```

```
dWfName=C10
```

```
dWfStepID=26
```

```
aliases=user20
```

DELETE_WORKFLOW

Description

This workflow related service deletes a named workflow.

- ❖ Given a named workflow, the service deletes the workflow.
- ❖ The most likely error is a named workflow not in the system. The service throws reasonable exceptions for display to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
dWfName	The workflow name.
IdcService	Must be set to DELETE_WORKFLOW.

Example

```
IdcService=DELETE_WORKFLOW
```

```
dWfName=C10
```

DELETE_WORKFLOWCRITERIA

Description

This workflow related service deletes a criteria workflow.

- ❖ Given a named workflow, the service deletes the workflow..
- ❖ This service is executed by the Workflow Admin applet. Criteria workflows and sub-workflows can be added, edited, enabled, disabled, and deleted from the *Criteria* tab of the Workflow Admin administration applet.
- ❖ The most likely error is a named workflow not in the system. The service throws reasonable exceptions for display to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
dWfName	The workflow name.
IdcService	Must be set to DELETE_WORKFLOWCRITERIA.

Example

```
IdcService=DELETE_WORKFLOWCRITERIA
```

```
dWfName=C10
```

DELETE_WORKFLOWDOCUMENTS

Description

This workflow related service deletes workflow content items.

- ❖ Given a named workflow and the content items to delete, the service deletes content from a basic workflow.
- ❖ This service is executed by the Workflow Admin applet. After adding a basic workflow, and adding content, the content is selected and **Delete** clicked to execute this service.
- ❖ The most likely error is a named workflow not in the system. The service throws reasonable exceptions for display to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
dWfName	The workflow name.
IdcService	Must be set to DELETE_WORKFLOWDOCUMENTS.

Example

```
IdcService=DELETE_WORKFLOWDOCUMENTS
```

```
dWfName=C10
```

```
docNames=1009,1008,1007
```

DELETE_WORKFLOWSTEP

Description

This workflow related service deletes a workflow step.

- ❖ Given a named workflow and a workflow step name, the service deletes the workflow step.
- ❖ The dWfStepID parameter is required for this service. The content server automatically assigns dWfStepID for that contribution stage.
- ❖ The most likely error is a named workflow or workflow step not in the system. If this service is unable to execute, this message is displayed to the user: *Unable to delete the step "{dWfStepName}" in the workflow "{dWfName}"*.

Required Parameters

These parameters must be specified:

Parameter	Description
dWfName	The workflow name.
dWfStepID	The workflow contribution stage ID. This auto-generated value can be retrieved from the database table.
dWfStepName	The workflow step name.
IdcService	Must be set to DELETE_WORKFLOWSTEP.

Example

```
IdcService=DELETE_WORKFLOWSTEP
```

```
dWfName=c2
```

IdcCommand Services

dWfStepID=12

dWfStepName=edit

DOC_FORMATS_WIZARD

Description

This service retrieves content item format extension.

- ❖ Used to load the content item configuration information and set the default content item format extension.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve file formats extension information.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to DOC_FORMATS_WIZARD.

Example

IdcService=DOC_FORMATS_WIZARD

DOC_INFO

Description

This service retrieves content item revision information.

- ❖ Given a content item revision ID, the service retrieves content item revision information (compare DOC_INFO_BY_NAME).
- ❖ The most likely errors are when the content item no longer exists in the system or when the user does not have the security level to perform this action. If this service is unable to execute, an error message is displayed to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
dID	The generated content item revision ID.
IdcService	Must be set to DOC_INFO.

Example

IdcService=DOC_INFO

dID=54321

DOC_INFO_BY_NAME

Description

This service retrieves content item revision information.

- ❖ Given a content item name, the service retrieves content item revision information (compare DOC_INFO).
- ❖ The most likely errors are when the content item no longer exists in the system or when the user does not have the security level to perform this action. If this service is unable to execute, an error message is displayed to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
dDocName	The content item identifier (Content ID).
IdcService	Must be set to DOC_INFO_BY_NAME.



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Example

```
IdcService=DOC_INFO_BY_NAME
dDocName=PublicDoc1_ia3c488971
```

DOC_INFO_LATESTRELEASE

Description

This Content Publisher related service retrieves the latest revision of a content item.

- ❖ Given a content item name, the service retrieves the latest revision.
- ❖ The most likely errors are when the content item no longer exists in the system or when the user does not have the security level to perform this action. If this service is unable to execute, an error message is displayed to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
dDocName	The content item identifier (Content ID).
IdcService	Must be set to DOC_INFO_LATESTRELEASE.



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Example

```
IdcService=DOC_INFO_LATESTRELEASE
dDocName=PublicDoc1_ia3c488971
```

DOC_SUBS_LIST

Description

This service returns a list of subscriptions.

- ❖ Given a user name, the service retrieves the subscription list for that user
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve subscriptions for "{dUser}".

Required Parameters

These parameters must be specified:

Parameter	Description
dSubscriptionID	The unique subscription ID.
dSubscriptionType	The subscription type.
IdcService	Must be set to DOC_SUBS_LIST.

Example

```
IdcService=DOC_SUBS_LIST  
dSubscriptionType=my_subscription  
dSubscriptionID=ADENG
```

DOWNLOAD_COMPONENT

Description

This Admin Server related service downloads a component.

- ❖ Executes a download of a component.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to download the existing component.

Required Parameters

These parameters must be specified:

Parameter	Description
name	The component zip file.
IdcService	Must be set to DOWNLOAD_COMPONENT.

Example

```
IdcService=DOWNLOAD_COMPONENT  
name=workflowdesigner.zip
```

EDIT_ALIAS

Description

This service edits the alias.

- ❖ Given an alias name, the service edits the alias. To edit and add users, the optional parameter `AliasUsersString` must be included.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to edit alias "{dAlias}".



Note: Editing the alias will delete previous data (Description and Users). To maintain previous data, you must provide old data and any additional data.

Required Parameters

These parameters must be specified:

Parameter	Description
<code>AliasUsersString</code>	The alias user or users. To add multiple users, use the <code>\n</code> escape sequence between each user name. For example: <code>AliasUsersString=sysadmin\nuser1</code>
<code>dAlias</code>	The alias name.
<code>dAliasDescription</code>	The alias description.
<code>IdcService</code>	Must be set to <code>EDIT_ALIAS</code> .

Example

```
IdcService=EDIT_ALIAS
```

```
dAlias=my_alias
```

IdcCommand Services

```
dAliasDescription=new user  
AliasUsersString=sysadmin\nuser1
```

EDIT_ARCHIVE

Description

This service edits an archive collection.

- ❖ Given content server instance name, the service edits the archive.
- ❖ The IDC_Name is normally the name of the collection. If you are using the default collection then this is the name of the content server instance.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to edit archive of collection "{InstanceMenuLabel}".



Caution: Using duplicate IDC_Name collection names will cause data corruption. The Archiver cannot be used to move or copy data between two instances that share the same IDC_Name. To do so will corrupt the data on the target system.

Required Parameters

These parameters must be specified:

Parameter	Description
IDC_Name	The name of the content server instance.
IdcService	Must be set to EDIT_ARCHIVE.

Example

```
IdcService=EDIT_ARCHIVE
IDC_Name=Master_on_server01
```

EDIT_ARCHIVEDATA

Description

This service edits archive data.

- ❖ Given content server instance name, the service edits the archive data.
- ❖ The IDC_Name is normally the name of the collection. If you are using the default collection then this is the name of the content server instance.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to edit archive of collection "{InstanceMenuLabel}".



Caution: Using duplicate IDC_Name collection names will cause data corruption. The Archiver cannot be used to move or copy data between two instances that share the same IDC_Name. To do so will corrupt the data on the target system.

Required Parameters

These parameters must be specified:

Parameter	Description
IDC_Name	The name of the content server instance.
IdcService	Must be set to EDIT_ARCHIVEDATA.

Example

```
IdcService=EDIT_ARCHIVEDATA
IDC_Name=Master_on_server01
```

EDIT_DOC_FORMATS

Description

This service edits content item formats.

- ❖ The service updates content item formats and file extension information.
- ❖ This service is executed in a browser interface by the Inbound Refinery. Select the Admin Links page, and click the File Formats Wizard link. When you check/uncheck any options and click Update, this service is called.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to update file formats or file extension information.

Required Parameters

These parameters must be specified:

Parameter	Description
dConversion	<p>The conversion algorithm is determined by the parameter dConversion.</p> <p>If the conversion type is not "PASSTHRU" or "NOCONVERSION" (default value) and the content is not web viewable or has a publish state, the content gets added to the queue to get converted. Otherwise, the server looks for the web-viewable (if applicable) and the vault file and updates the processing state to say that the file has already been converted.</p> <p>If the file mime-type is Application/FDF, the server will set the conversion to "EXCHANGE-FDF"..</p>
dDescription	The provided description.

Parameter	Description
dFormat	The formatting process used to create the web viewable version of the content. For example, application/hcsf or application/doc.
extensions	The file extension such as DOC, TXT, PDF.
IdcService	Must be set to EDIT_DOC_FORMATS.

Example

```
IdcService=EDIT_DOC_FORMATS  
dFormat=application/msword  
extensions=doc  
dConversion=PASSTHRU  
dDescription=adding the description
```

EDIT_DOCEXTENSION

Description

This service edits the file extension.

- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to edit file extension.

Required Parameters

These parameters must be specified:

Parameter	Description
dExtension	The file extension such as HCSF, DOC, TXT.
dFormat	The formatting process used to create the web viewable version of the content. For example, application/hcsf or application/doc.
IdcService	Must be set to EDIT_DOCEXTENSION.

Example

```
IdcService=EDIT_DOCEXTENSION
```

```
dExtension=hcsf
```

```
dFormat=application/hcsf
```

EDIT_DOCFORMAT

Description

This service updates the content item format.

- ❖ This service is called from Configuration Manager applet.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to edit file format.

Required Parameters

These parameters must be specified:

Parameter	Description
dConversion	The conversion algorithm is determined by the parameter dConversion. If the conversion type is not "PASSTHRU" or "NOCONVERSION" (default value) and the content is not web viewable or has a publish state, the content gets added to the queue to get converted. Otherwise, the server looks for the web-viewable (if applicable) and the vault file and updates the processing state to say that the file has already been converted. If the file mime-type is Application/FDF, the server will set the conversion to "EXCHANGE-FDF".
dDescription	The provided description.
dFormat	The formatting process used to create the web viewable version of the content. For example, application/hcsf or application/doc.
IdcService	Must be set to EDIT_DOCFORMAT

Example

```
IdcService=EDIT_DOCFORMAT  
dFormat=application/hcsf  
dDescription=hypercontent  
dConversion=PASSTHRU
```

EDIT_DOCTYPE

Description

This service edits the file type.

- ❖ Given a named content item type, the service updates the content item format type.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to edit file format "{dDocType}".

Required Parameters

These parameters must be specified:

Parameter	Description
dDescription	The provided description.
dDocType	The content item type.
dGif	The name of the GIF image with file extension.
IdcService	Must be set to EDIT_DOCTYPE.

Example

```
IdcService=EDIT_DOCTYPE
dDocType=MY_TEST
dDescription=edit testing
dGif=admkt.gif
```

EDIT_EXPORTERS

Description

This service registers exporters for an archive.

- ❖ Given an archive name and content server instance name, the service registers exporters for the archive.
- ❖ The IDC_Name is normally the name of the collection. If you are using the default collection then this is the name of the content server instance.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to register exporters for the archive "{aArchiveName}" of "{InstanceMenuLabel}".



Caution: Using duplicate IDC_Name collection names will cause data corruption. The Archiver cannot be used to move or copy data between two instances that share the same IDC_Name. To do so will corrupt the data on the target system.

Required Parameters

These parameters must be specified:

Parameter	Description
aArchiveName	The archive name.
IDC_Name	The name of the content server instance.
IdcService	Must be set to EDIT_EXPORTERS

Example

```
IdcService=EDIT_EXPORTERS
IDC_Name=Master_on_server01
```

IdcCommand Services

```
aArchiveName=archive_test
```

EDIT_GROUP

Description

This service edits a group.

- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to edit group "{dGroupName}".



Important: The value `dPrivilege` defines the privilege setting. If this value is set to 0 only admin has RWDA privilege on the group, no other roles will be able to access the group. If the value is set to 15 then all the roles get RWDA privilege on the group. For more information about numbering of privilege, please refer to the *Content Server Security Guide*.

Required Parameters

These parameters must be specified:

Parameter	Description
<code>dDescription</code>	The provided description.
<code>dGroupName</code>	The defined group name.
<code>dPrivilege</code>	The privilege setting.
<code>IdcService</code>	Must be set to <code>EDIT_GROUP</code> .

Example

```
IdcService=EDIT_GROUP
dGroupName=MY_GROUP
dPrivilege=0
dDescription=admin privileges
```

EDIT_METADEF

Description

This service updates the meta data information in the meta definition field.

- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to edit meta definition field.

Required Parameters

These parameters must be specified:

Parameter	Description
dCaption	The caption.
dDefaultValue	The defined default value.
dIsEnabled	Defines whether this field is enabled. <ul style="list-style-type: none"> ❖ Set to TRUE (1) to enable. ❖ Set to FALSE (0) to disable.
dIsOptionList	Defines whether the metadata (information field) is an option list. <ul style="list-style-type: none"> ❖ Set to TRUE (1) for an option list. For example, this may be used for a custom metadata option list for departments or divisions. ❖ Set to FALSE (0) if not an option list. For example, this may be used for a custom metadata field for comments or policy number.
dIsRequired	Defines whether this is a required field. <ul style="list-style-type: none"> ❖ Set to TRUE (1) to require this field. ❖ Set to FALSE (0) to make this field optional.

Parameter	Description
dIsSearchable	Defines whether this is a searchable field. <ul style="list-style-type: none"> ❖ Set to TRUE (1) to enable this field as searchable. ❖ Set to FALSE (0) to make this field not searchable.
dName	The unique name for the metadata (information field). For example, you could define custom metadata fields such as xApproved_by, xComments, xDepartment, xDivision, xReviewDate.
dOptionListKey	The option metadata field list key.
dOptionListType	The option metadata field list type.
dOrder	The defined order (numeric data type).
dType	The metadata (information field) type such as Text, BigText, Memo, Date.
IdcService	Must be set to EDIT_METADEF

Example

```

IdcService=EDIT_METADEF
dName=xDepartment
dType=BigText
dIsRequired=1
dIsEnabled=1
dIsSearchable=1
dCaption=Caption_Changed
dIsOptionList=1
dDefaultValue=two
dOptionListKey=xDepartmen_fieldList

```

IdcCommand Services

```
dOptionListType=chunval  
dOrder=4
```

EDIT_ROLE

Description

This service edits a role.

- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to edit role "{dRoleName}".



Important: The value `dPrivilege` defines the privilege setting. If this value is set to 0 only admin has RWDA privilege on the group, no other roles will be able to access the group. If the value is set to 15 then all the roles get RWDA privilege on the group. For more information about numbering of privilege, please refer to the *Content Server Security Guide*.

Required Parameters

These parameters must be specified:

Parameter	Description
<code>dGroupName</code>	The defined group name.
<code>dPrivilege</code>	The privilege setting.
<code>dRoleName</code>	The unique role name.
<code>IdcService</code>	Must be set to <code>EDIT_METADEF</code> .

Example

```
IdcService=EDIT_ROLE
dRoleName=test_role
dGroupName=MY_GROUP
dPrivilege=15
```

EDIT_USER

Description

This service edits an existing user.

- ❖ Given a user name and user authorization type, the service determines if the user is in the system. If the user does not exist, the service fails. Otherwise the user information is updated and replaced.
- ❖ The most likely error is the user not having the security level to perform this action. If this service is unable to execute, an error message is displayed to the user.



Note: The user attribute information replaces the current attributes. It does not add to the list. Consequently, if the user attribute information is not defined, the user will become a guest in the system.

Required Parameters

These parameters must be specified:

Parameter	Description
dName	The unique name.
dUserAuthType	The user authorization type. This value must be set to either LOCAL or GLOBAL.
IdcService	Must be set to EDIT_USER.

Optional Parameters

These optional parameters may be specified:

Parameter	Description
dEmail	The email address of the user.
dFullName	The full name of the user.
dPassword	The password for the user.
dUserLocale	The locale designation such as English-US, English-UK, Deutsch, Français, Español.
dUserType	The defined user type.

Optional Attribute Information

A result set containing the user's attribute information and referencing the roles the user belongs to and the accounts the user has access to. Attribute information consists of a list of three comma-separated strings. The first string indicates the type of attribute, the second the name of the attribute, and the third is the access number for *accounts* or default entry for *role*.



Important: The user attribute information is not pre-defined. The user by default will belong to no roles or accounts, and will become a guest in the system.

Attribute Information	Description
Access Number	The access number determines the level of access or privileges assigned to the user
Attribute Name	The attribute name is the name of the <i>role</i> or <i>account</i> to be assigned. For example, admin, contributor, or editor may be assigned.

Attribute Information	Description
Attribute Type	The attribute types consist of <i>role</i> or <i>account</i> .

Access Number

These access numbers can be assigned to the user.

Access Level Flags	Description
1	Read only.
3	Read and write.
7	Read, write, and delete.
15	Administrative privileges.

Attribute Name

A user can belong to multiple roles and accounts, there may be multiple role and account information strings separated by commas in the attribute information column.

- ❖ If the user is to have the admin role, define the user attribute information as follows:

```
@ResultSet UserAttribInfo
2
dUserName
AttributeInfo
jsmith
role,contributor,15
@end
```

- ❖ If the user is to belong to both the contributor and editor roles and have read privilege on the account books, define the user attribute information as:

```

@ResultSet UserAttribInfo
2
dUserName
AttributeInfo
jsmith
role,contributor,15,role,editor,15,account,books,1
@end

```

Attribute Type

When defining a role, the first string specifies that this is a role attribute, the second string is the name of the role, and the third is the default entry of 15.

When defining an account, the first string specifies that this is an account attribute, the second string is the name of the account, and the third is the access level.

- ❖ For an attribute role, the information is in the form:

```
role,contributor,15
```

- ❖ For an attribute account where the access level determines the users rights to the named account, the information is in the form:

```
account,books,1
```

Example

- ❖ IdcCommand command file format:

```

IdcService=EDIT_USER
dName=user20
dUserAuthType=Local

```

- ❖ Serialized HDA format with optional parameters and attribute information:

```
@Properties LocalData
```

IdcCommand Services

```
IdcService=EDIT_USER
dName=jsmith
dFullName=Jennifer Smith
dUserAuthType=Local
dPassword=password
dEmail=jsmith@stellent.com
dUserType=MKT
dUserLocale=English-US
@end
```

```
@ResultSet UserAttribInfo
2
dUserName
AttributeInfo
jsmith
role,admin,15,role,contributor,15
@end
```

EDIT_USER_PROFILE

Description

This service edits an existing user profile.

- ❖ Given a user name, the service edits the user profile.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to edit user "{dName}".

Required Parameters

These parameters must be specified:

Parameter	Description
dName	The unique name.
IdcService	Must be set to EDIT_USER_PROFILE.

Optional Parameters

These optional parameters may be specified:

Parameter	Description
dEmail	The email address of the user.
dFullName	The full name of the user.
dPassword	The password for the user.
dUserLocale	The locale designation such as English-US, English-UK, Deutsch, Français, Español.
dUserType	The defined user type.

Example

- ❖ IdcCommand command file format:

```
IdcService=EDIT_USER_PROFILE  
dName=sysadmin
```

- ❖ Serialized HDA format with optional parameters:

```
@Properties LocalData  
IdcService=EDIT_USER_PROFILE  
dName=jsmith  
dFullName=Jennifer Smith  
dPassword=password  
dEmail=jsmith@stellent.com  
dUserType=MKT  
dUserLocale=English-US  
@end
```

EDIT_WF_TEMPLATE

Description

This workflow related service edits a template.

- ❖ These parameters can be executed in a browser environment using the Workflow Admin applet. For example, the `dWfStepName`, `dWfStepDescription`, and `dWfStepType` entries correspond to the *Name* entry, the *Description* field, and the *Type* drop-down list of the Workflow Admin applet Edit Step Reviewer screen.
- ❖ Setting `dWfStepHasWeight` to TRUE (1) is the same as enabling the “At least this many reviewers” option for Workflow Steps in the Workflow Admin applet. If Set to TRUE, the parameter `dWfStepWeight` should be assigned a numeric value and `dWfStepIsAll` should be set to FALSE.
- ❖ Setting `dWfStepWeight` to a numeric value is the same as entering a value in the field associated with the “At least this many reviewers” option for Workflow Steps in the Workflow Admin applet. If defined, the workflow passes to the next step as soon as the number of users specified have approved the revision.
- ❖ Setting `dWfStepIsAll` to TRUE (1) is the same as enabling the “All Reviewers” option for Workflow Steps in the Workflow Admin applet. If set to TRUE, all users assigned to the step must approve the revision before the workflow passes to the next step.
- ❖ If this service is unable to execute, this message is displayed to the user: *Unable to edit the workflow template "{dWfTemplateName}"*.

Required Parameters

These parameters must be specified:

Parameter	Description
dAliases	The list of alias users separated by the /n escape sequence. For example: dAliases=sysadmin user/nservices use
dWfStepDescription	The provided step description.
dWfStepHasWeight	Enables the limited reviewer option. <ul style="list-style-type: none"> ❖ Set to TRUE (1), and define the number of reviewers with the dWfStepWeight parameter. ❖ Set to FALSE (0) if dWfStepIsAll is enabled.
dWfStepIsAll	Enables the all reviewer option. <ul style="list-style-type: none"> ❖ Set to TRUE (1) to require all users assigned to the step to approve the revision before the workflow passes to the next step. ❖ Set to FALSE (0) if dWfStepHasWeight is enabled.
dWfStepName	The workflow step name. Typically this is the type of review (such as <i>initial review</i> or <i>copy edit</i>) or the function of the reviewer (such as <i>manager</i> or <i>copy editor</i>).
dWfStepType	The workflow step type: <ul style="list-style-type: none"> ❖ Reviewer—Approves or rejects the revision. ❖ Reviewer/Contributor—Can edit the revision and approves or rejects it.

Parameter	Description
dWfStepWeight	<p>Defines the number of reviewers for the limited reviewer option.</p> <ul style="list-style-type: none"> ❖ Enter a numeric value for the number of reviewers. The workflow passes to the next step as soon as the number of users specified have approved the revision. ❖ You can enter zero (0) in this field to notify reviewers that the revision has reached the step, but reviewers will not be able to approve, reject, or edit the revision at that step. The workflow will pass to the next step automatically.
dWfTemplateDescription	The provided workflow template description.
dWfTemplateName	The workflow template name.
IdcService	Must be set to EDIT_WF_TEMPLATE.

Example

```

IdcService=EDIT_WF_TEMPLATE
dWfTemplateName=New_Template
dWfTemplateDescription=departmental template
dWfStepName=edit
dWfStepType=Reviewer/Contribution
dWfStepIsAll=0
dWfStepWeight=2
dWfStepDescription=adding another step
dAliases=sysadmin user/nservices user
dWfStepHasWeight=1

```

EDIT_WORKFLOW

Description

This workflow related service edits a workflow.

- ❖ Given a workflow name, the service updates the workflow.
- ❖ This service is executed by the Workflow Admin applet.
- ❖ Workflows and sub-workflows can be added, edited, enabled, disabled, and deleted from the Workflow Admin administration applet.
- ❖ The most likely error is a workflow name that does not exist or a user failing the security check. The service throws reasonable exceptions for display to the user in these situations.

Required Parameters

These parameters must be specified:

Parameter	Description
dWfName	The workflow name.
IdcService	Must be set to EDIT_WORKFLOW.

Optional Parameters

These optional parameters may be specified:

Parameter	Description
dSecurityGroup	The security group such as PUBLIC or SECURE.
dWfCriteriaName	The workflow criteria name.
dWfCriteriaOperator	The workflow criteria operator such as MATCHES, STARTS, ENDS

Parameter	Description
dWfCriteriaValue	The workflow criteria value.
dWfDescription	The workflow description.
dWfType	The workflow type such as BASIC or CRITERIA.



Note: Refer to the Content Publisher documentation for additional information on workflow types and workflow criteria.

Example

- ❖ IdcCommand command file format:

```
IdcService=EDIT_WORKFLOW
dWfName=c2
```

- ❖ Serialized HDA format with optional parameters:

```
@Properties LocalData
IdcService=EDIT_WORKFLOW
dWfName=test_workflow
dSecurityGroup=Public
dWfType=Criteria
dWfCriteriaName=dDocAuthor
dWfCriteriaOperator=matches
dWfCriteriaValue=sysadmin
dWfDescription=testing workflow
@end
```

EDIT_WORKFLOWCRITERIA

Description

This workflow related service edits workflow criteria.

- ❖ This service is executed by the Workflow Admin applet.
- ❖ Criteria workflows and sub-workflows can be added, edited, enabled, disabled, and deleted from the *Criteria* tab of the Workflow Admin administration applet.
- ❖ The most likely error is a workflow name that does not exist or a user failing the security check. The service throws reasonable exceptions for display to the user in these situations.

Required Parameters

These parameters must be specified:

Parameter	Description
dSecurityGroup	The security group such as PUBLIC or SECURE.
dWfCriteriaName	The workflow criteria name.
dWfCriteriaOperator	The workflow criteria operator such as MATCHES, STARTS, ENDS
dWfCriteriaValue	The workflow criteria value.
dWfName	The workflow name.
IdcService	Must be set to EDIT_WORKFLOWCRITERIA.

Example

```
IdcService=EDIT_WORKFLOWCRITERIA
```

```
dWfName=c2
```

```
dSecurityGroup=Public  
dWfCriteriaName=dDocType  
dWfCriteriaOperator=matches  
dWfCriteriaValue=ADACCT
```

EDIT_WORKFLOWSTEP

Description

This workflow related service edits a workflow step.

- ❖ These parameters can be executed in a browser environment using the Workflow Admin applet. For example, the `dWfStepName`, `dWfStepDescription`, and `dWfStepType` entries correspond to the *Name* entry, the *Description* field, and the *Type* drop-down list of the Workflow Admin applet Edit Step Reviewer screen.
- ❖ Setting `dWfStepWeight` to a numeric value is the same as entering a value in the field associated with the “At least this many reviewers” option for Workflow Steps in the Workflow Admin applet. If defined, the workflow passes to the next step as soon as the number of users specified in `dWfStepWeight` have approved the revision.
- ❖ Setting `dWfStepIsAll` to `TRUE` (1) is the same as enabling the “All Reviewers” option for Workflow Steps in the Workflow Admin applet. If set to `TRUE`, all users assigned to the step must approve the revision before the workflow passes to the next step.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to edit the step "{dWfStepName}" in the workflow "{dWfName}".

Required Parameters

These parameters must be specified:

Parameter	Description
<code>dAlias</code>	The alias name.
<code>dAliasType</code>	The assigned alias type.
<code>dWfName</code>	The workflow name.

Parameter	Description
dWfStepDescription	The provided step description.
dWfStepIsAll	<p>Enables the all reviewer option.</p> <ul style="list-style-type: none"> ❖ Set to TRUE (1) to require all users assigned to the step to approve the revision before the workflow passes to the next step. ❖ Set to FALSE (0) to enable the dWfStepWeight limited reviewer option.
dWfStepName	The workflow step name. Typically this is the type of review (such as <i>initial review</i> or <i>copy edit</i>) or the function of the reviewer (such as <i>manager</i> or <i>copy editor</i>).
dWfStepType	<p>The workflow step type:</p> <ul style="list-style-type: none"> ❖ Reviewer—Approves or rejects the revision. ❖ Reviewer/Contributor—Can edit the revision and approves or rejects it.

Parameter	Description
dWfStepWeight	<p>Defines the number of reviewers for the limited reviewer option.</p> <ul style="list-style-type: none"> ❖ Enter a numeric value for the number of reviewers. The workflow passes to the next step as soon as the number of users specified have approved the revision. ❖ You can enter zero (0) in this field to notify reviewers that the revision has reached the step, but reviewers will not be able to approve, reject, or edit the revision at that step. The workflow will pass to the next step automatically. ❖ The parameter dWfStepIsAll is initially evaluated. If dWfStepIsAll is set to FALSE, this option is enabled.
IdcService	Must be set to EDIT_WORKFLOWSTEP

Optional Parameters

These parameters may be specified:

Parameter	Description
wfEntryScript	The programmatic entry script.
wfExitScript	The programmatic exit script.
wfUpdateScript	The programmatic update script.



Note: Programmatic scripts must be placed within <\$ and \$> delimiters. For example, <\$if ConditionExpression\$>.

Example

```
IdcService=EDIT_WORKFLOWSTEP
dWfName=test_workflow
dWfStepName=edit
dWfStepType=Reviewer/Contributor
dWfStepDescription=copy edit step
dAliasType=user
dAlias=user20
dWfStepIsAll=0
dWfStepWeight=2
wfEntryScript=
wfExitScript=
wfUpdateScript=
```

ENABLE_DISABLE_PROVIDER

Description

This service changes the provider state.

- ❖ Given a provider name and type, the service reverses the provider enable state. If the provider is enabled, the service disables the provider. If the provider is disabled, the service enables the provider.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to change provider "{pName}" enable state.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to ENABLE_DISABLE_PROVIDER.
pName	The provider name.

Parameter	Description
pType	<p>The provider type:</p> <ul style="list-style-type: none"> ❖ Database—An information repository server that provides an API for connecting and communicating with it. This retrieves information and enables information to be changed in the database. Examples of this type are system databases and LDAP. ❖ Incoming—A connection initiated from an outside entity like a browser or client application. The server listens and is aware of incoming connections ❖ Outgoing—A connection initiated to an outside entity. You can use this type to communicate between Content Server instances. ❖ Preview—The API that establishes connections between content server and Preview technology like the DTM server for HTML preview and Content Categorizer.

Example

```
IdcService=ENABLE_DISABLE_PROVIDER
pName=Proxied_2_on_test13
pType=outgoing
```

EXPORT_ARCHIVE

Description

This service performs an export of a specified archive.

- ❖ Given a collection and an archive in the collection, this service will export the archive. The export will read in the archive's definition file and will follow the export rules specified. This includes specifying the export query, whether or not users and/or document configuration information should be exported, and whether or not previous batch files should be deleted.
- ❖ The server executes this service asynchronously. This means that the server needs to be running for the services to be even executed and that IdcCommand will exit announcing success, when actually all it has done is told the server to perform the action.
- ❖ The server can only export or import archives one at a time. A batch file that has multiple exports and imports will declare success on the first and failure on all subsequent commands until it has finished the first. This means that a batch file should have only one export or import in it and some external process will need to determine that the action has completed before issuing another export/import command.
- ❖ If the parameter *aDoDelete* is set to TRUE (1), the revision will be deleted after successful archive.
- ❖ The parameter *dataSource* should always be set to *RevisionIDs*. This is the query stub which along with the export query will be used to create the list of revisions to export.
- ❖ The *IDC_Name* is normally the name of the collection. If you are using the default collection then this is the name of the content server instance.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to export archive "{aArchiveName}".



Caution: Using duplicate IDC_Name collection names will cause data corruption. The Archiver cannot be used to move or copy data between two instances that share the same IDC_Name. To do so will corrupt the data on the target system.

Required Parameters

These parameters must be specified:

Parameter	Description
aArchiveName	The archive name.
dataSource	Must be set to RevisionIDs.
IDC_Name	The collection name or content server instance the collection belongs to.
IdcService	Must be set to EXPORT_ARCHIVE.

Optional Parameters

This parameter may be specified:

Parameter	Description
aDoDelete	Set to TRUE (1), to delete the successfully archived revisions.

Example

- ❖ IdcCommand command file format (exports the archive “archive_test” in the collection “Master_on_server01”):

```
# To export an archive
IdcService=EXPORT_ARCHIVE
aArchiveName=archive_test
```

IdcCommand Services

```
IDC_Name=Master_on_server01  
dataSource=RevisionIDs
```

❖ Serialized HDA format with optional parameter:

```
@Properties LocalData  
IdcService=EXPORT_ARCHIVE  
aArchiveName=archive_test  
IDC_Name=Master_on_server01  
dataSource=RevisionIDs  
aDoDelete=1  
@end
```

FORM_PROCESS

Description

This service processes a form for submission.

- ❖ This service is called by FORM_SUBMIT to process PDF forms. It uses the auto-generated *ClientId* parameter.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to complete form submission.
- ❖ See also FORM_SUBMIT.



Note: This service is used to submit PDF forms and cannot be executed by IdcCommand.

Example

N/A

FORM_SUBMIT

Description

This service submits PDF forms.

- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to begin form submission.
- ❖ See also FORM_PROCESS



Note: This service is used to submit PDF forms and cannot be executed by IdcCommand.

Example

N/A

GET_ACTIVE_WORKFLOWS

Description

This workflow related service returns the workflow list.

- ❖ The service retrieves the workflow list and displays all active workflows.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve workflow list.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_ACTIVE_WORKFLOWS.

Example

```
IdcService=GET_ACTIVE_WORKFLOWS
```

GET_ADD_EDIT_PROVIDER_FORM

Description

This form return related service updates the provider form.

- ❖ This service creates a provider form if one does not exist or updates an existing provider form.
- ❖ This service executes from the browser interface when adding or editing a provider.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to create provider form.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_ADD_EDIT_PROVIDER_FORM.
isEdit	Defines the add/edit function. <ul style="list-style-type: none"> ❖ Set to TRUE if editing a provider ❖ Set to FALSE if adding a provider.

Parameter	Description
pType	<p>The provider type:</p> <ul style="list-style-type: none"> ❖ Database—An information repository server that provides an API for connecting and communicating with it. This retrieves information and enables information to be changed in the database. Examples of this type are system databases and LDAP. ❖ Incoming—A connection initiated from an outside entity like a browser or client application. The server listens and is aware of incoming connections ❖ Outgoing—A connection initiated to an outside entity. You can use this type to communicate between Content Server instances. ❖ Preview—The API that establishes connections between content server and Preview technology like the DTM server for HTML preview and Content Categorizer.

Additional Parameters

These parameters may be required:

Parameter	Description
pName	The provider name. Required when editing a provider.
ResourceTemplate	The resource template. Required when adding a provider.

Example

- ❖ If you are adding a provider, the parameters you pass look like the following:

```
IdcService=GET_ADD_EDIT_PROVIDER_FORM  
pType=preview  
ResourceTemplate=PROVIDER_ADD_PREVIEW  
isEdit=0
```

- ❖ If you are editing a provider then the parameters you pass look like the following:

```
IdcService=GET_ADD_EDIT_PROVIDER_FORM  
pType=outgoing  
pName=Proxied_2_on_server02  
isEdit=1
```

GET_ADMIN_PAGE

Description

This service returns the administrator page.

- ❖ This action executes an HTML page request for the administrator page. If the user is not assigned the administrator or sub administrator role, the service returns an error message.
- ❖ If this service is unable to execute, this message is displayed to the user: *Unable to retrieve referenced HTML page.*
- ❖ This service requires the name of the page to retrieve and the action to execute.

Required Parameters

These parameters must be specified:

Parameter	Description
Action	The action to execute. Usually set to <i>GetTemplatePage</i> .
IdcService	Must be set to GET_ADMIN_PAGE.
Page	The HTML page requested.

Example

To get the admin links page the required parameters will be:

```
IdcService=GET_ADMIN_PAGE
```

```
Page=ADMIN_LINKS
```

```
Action=GetTemplatePage
```

GET_ALIASES

Description

This service retrieves the alias list containing user alias information.

- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve alias list.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_ALIASES.

Example

IdcService=GET_ALIASES

GET_ALL_PROVIDERS

Description

This service retrieves the provider list containing all providers.

- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve provider list.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_ALL_PROVIDERS

Example

```
IdcService=GET_ALL_PROVIDERS
```

GET_ARCHIVECOLLECTIONS

Description

This service retrieves the list of archive collections.

- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve the list of collections.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_ARCHIVECOLLECTIONS.

Example

```
IdcService=GET_ARCHIVECOLLECTION
```

GET_ARCHIVED_FILE

Description

This service returns a specific rendition of an archived content item.

- ❖ The GET_ARCHIVED_FILE command computes the content item revision ID (*dID*) for the revision, and then determines the filename of a particular rendition of the revision with the computed content item revision ID (*dID*). A specified content item revision ID (*dID*) or a content item name (*dDocName*) along with a *RevisionSelectionMethod* parameter can be used.
- ❖ Given a *dID* or a *dDocName* along with a *RevisionSelectionMethod* parameter, the service determines the filename of a particular rendition of the revision and returns that file to the client.
- ❖ The most likely errors are mismatched parameters or a request for a revision or rendition that does not exist. The service throws reasonable exceptions for display to the user in these situations.



Note: It is recommended that *dDocName* be present in all requests for content items where the requester knows the *dDocName*. Error messages in the content server assume that it is present, as do other features such as forms.

Required Parameters



Important: Either the content item revision ID (*dID*) must be specified or a content item name (*dDocName*) along with a *RevisionSelectionMethod* parameter must be defined.

Parameter	Description
IdcService	Must be set to GET_ARCHIVED_FILE.

Optional Parameters

These optional parameters may be specified:

Parameter	Description
dDocName	<p>The content item identifier (Content ID).</p> <ul style="list-style-type: none"> ❖ If dDocName is not present, dID must be present and RevisionSelectionMethod must not be present. ❖ If RevisionSelectionMethod is present, a rendition of a revision of the content item with this name will be returned, if it exists. If RevisionSelectionMethod is not present, dDocName may be used in error messages.
dID	<p>The generated content item revision ID.</p> <ul style="list-style-type: none"> ❖ If dID is not specified, dDocName and RevisionSelectionMethod must be specified. ❖ A rendition of the revision of the content item with this ID will be returned, if it exists, and the RevisionSelectionMethod parameter does not exist or has the value <i>Specific</i>.
Rendition	<p>The content item rendition. This parameter specifies the rendition of the content item and can be set to <i>Primary</i>, <i>Web</i>, or <i>Alternate</i>. If Rendition is not present, it defaults to <i>Primary</i>.</p> <ul style="list-style-type: none"> ❖ If the value is <i>Primary</i>, the primary rendition of the selected revision is returned. ❖ If the value is <i>Web</i>, the web viewable rendition of the selected revision is returned. ❖ If the value is <i>Alternate</i>, the alternate rendition of the selected revision is returned.

Parameter	Description
RevisionSelectionMethod	<p>The revision selection method.</p> <ul style="list-style-type: none"> ❖ If present, dDocName must be present. The value of this variable is the method used to compute a dID from the specified dDocName. Its value may be <i>Specific</i>, <i>Latest</i>, or <i>LatestReleased</i>. ❖ If the value is <i>Specific</i>, the dDocName is ignored, and dID is required and is used to get a rendition. If the value is <i>Latest</i>, the latest revision of the content item is used to compute the dID. If the value is <i>LatestReleased</i>, the latest released revision of the content item is used to compute the dID.



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Example

```
IdcService=GET_ARCHIVED_FILE
dDocName=notice
RevisionSelectionMethod=LatestReleased
Rendition=web
```

GET_ARCHIVES

Description

This service retrieves a list of archives.

- ❖ Given a content server instance, the service returns the list of archives in that collection.
- ❖ The IDC_Name is normally the name of the collection. If you are using the default collection then this is the name of the content server instance.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve the list of archives for the collection
"{InstanceMenuLabel}"



Caution: Using duplicate IDC_Name collection names will cause data corruption. The Archiver cannot be used to move or copy data between two instances that share the same IDC_Name. To do so will corrupt the data on the target system.

Required Parameters

These parameters must be specified:

Parameter	Description
IDC_Name	The name of the content server instance.
IdcService	Must be set to GET_ARCHIVES.

Example

```
IdcService=GET_ARCHIVES
IDC_Name=Master_on_server01
```

GET_BATCH_FILE_DOCUMENTS

Description

This service retrieves all batch file content items.

- ❖ The IDC_Name is normally the name of the collection. If you are using the default collection then this is the name of the content server instance.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve batch file content item.



Caution: Using duplicate IDC_Name collection names will cause data corruption. The Archiver cannot be used to move or copy data between two instances that share the same IDC_Name. To do so will corrupt the data on the target system.

Required Parameters

These parameters must be specified:

Parameter	Description
aArchiveName	The archive name.
aBatchFile	The HDA batch file.
IDC_Name	The name of the content server instance.
IdcService	Must be set to GET_BATCH_FILE_DOCUMENTS.

Example

```
IdcService=GET_BATCH_FILE_DOCUMENTS
IDC_Name=Master_on_server01
aArchiveName=latest_archive
```

IdcCommand Services

```
aBatchFile=02-jan-18_09.34.41_430/0218093441~1.hda
```

GET_BATCH_SCHEMA

Description

This service retrieves the batch schema.

- ❖ Schema information describes the classes of objects that are stored in the database.
- ❖ The IDC_Name is normally the name of the collection. If you are using the default collection then this is the name of the content server instance
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve the batch schema.



Caution: Using duplicate IDC_Name collection names will cause data corruption. The Archiver cannot be used to move or copy data between two instances that share the same IDC_Name. To do so will corrupt the data on the target system.

Required Parameters

These parameters must be specified:

Parameter	Description
aArchiveName	The archive name.
aBatchFile	The HDA batch file.
IDC_Name	The name of the content server instance.
IdcService	Must be set to GET_BATCH_FILE_SCHEMA.

Example

```
IdcService=GET_BATCH_SCHEMA
IDC_Name=Master_on_server01
```

IdcCommand Services

aArchiveName=latest_archive

aBatchFile=02-jan-18_09.34.41_430/0218093441~1.hda

GET_BATCH_VALUES

Description

This service retrieves the metadata field values for the batch.

- ❖ The IDC_Name is normally the name of the collection. If you are using the default collection then this is the name of the content server instance
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve the batch values.



Caution: Using duplicate IDC_Name collection names will cause data corruption. The Archiver cannot be used to move or copy data between two instances that share the same IDC_Name. To do so will corrupt the data on the target system.

Required Parameters

These parameters must be specified:

Parameter	Description
aArchiveName	The archive name.
aBatchFile	The HDA batch file.
IDC_Name	The name of the content server instance.
IdcService	Must be set to GET_BATCH_VALUES.

Example

```
IdcService=GET_BATCH_VALUES
IDC_Name=Master_on_server01
aArchiveName=latest_archive
```

IdcCommand Services

```
aBatchFile=02-jan-18_09.34.41_430/0218093441~1.hda
```

GET_BATCHFILES

Description

This service retrieves batch files.

- ❖ Given an instance menu label, the service retrieves the associated batch files.
- ❖ The IDC_Name is normally the name of the collection. If you are using the default collection then this is the name of the content server instance.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve the batch files for "{InstanceMenuLabel}".



Caution: Using duplicate IDC_Name collection names will cause data corruption. The Archiver cannot be used to move or copy data between two instances that share the same IDC_Name. To do so will corrupt the data on the target system..

Required Parameters

These parameters must be specified:

Parameter	Description
aArchiveName	The archive name.
IDC_Name	The name of the content server instance.
IdcService	Must be set to GET_BATCHFILES.

Example

```
IdcService=GET_BATCHFILES
```

```
IDC_Name=Master_on_server01
```

IdcCommand Services

aArchiveName=latest_archive

GET_CRITERIA_WORKFLOWS_FOR_GROUPS

Description

This workflow related service returns criteria workflow information.

- ❖ Given a named security group, this service returns a list of workflows and related steps.
- ❖ Returns the result sets *WorkflowsForGroup* and *WorkflowStepsForGroup*:
 - *WorkflowsForGroup* lists all of the workflows for this group (dWfID, dWfName).
 - *WorkflowStepsForGroup* lists all of the steps in all of the workflows for this group (dWfID, dWfName, dWfStepID, dWfStepName).
- ❖ Criteria workflows and sub-workflows can be added, edited, enabled, disabled, and deleted from the *Criteria* tab of the Workflow Admin administration applet.
- ❖ The most likely error is a named security group that does not exist or a user failing the security check. The service throws reasonable exceptions for display to the user in these situations.

Required Parameters

These parameters must be specified:

Parameter	Description
dSecurityGroup	The security group such as PUBLIC or SECURE.
IdcService	Must be set to GET_CRITERIA_WORKFLOWS_FOR_GROUPS.

Example

- ❖ IdcCommand command file format:

```
# Retrieves criteria workflow information
IdcService=GET_CRITERIA_WORKFLOWS_FOR_GROUPS
dSecurityGroup=Public
```

- ❖ Serialized HDA format:

```
@Properties LocalData
IdcService=GET_CRITERIA_WORKFLOWS_FOR_GROUPS
dSecurityGroup=Public
@end
```

GET_DATARESULTSET

Description

This service retrieves result set information.

- ❖ Given a defined data source, the service retrieves data result set information.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve information about {dataSource}.

Required Parameters

These parameters must be specified:

Parameter	Description
dataSource	The data source.
IdcService	Must be set to GET_DATARESULTSET.
whereClause	The programmatic clause.

Example

```
IdcService=GET_DATARESULTSET
dataSource=Users
whereClause=((dUserAuthType LIKE 'LOCAL' ))
```

GET_DOC_CONFIG_INFO

Description

This Content Publisher related service retrieves configuration information.

- ❖ This service returns all the configuration information. It does not need any other parameters.
- ❖ Returns the properties information, the security group option list, and these result sets:
 - DOC_DEFAULT_INFO
 - DocMetaDefinition
 - DocTypes
 - DocFormats
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve configuration information for content item check in.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_DOC_CONFIG_INFO.

Example

- ❖ IdcCommand command file format:


```
# Retrieves content item configuration information
IdcService=GET_DOC_CONFIG_INFO
```
- ❖ Serialized HDA format:

```
@Properties LocalData  
IdcService=GET_DOC_CONFIG_INFO  
@end
```

GET_DOC_PAGE

Description

This service executes an HTML page request.

- ❖ This service is usually called from the browser interface.
- ❖ This service requires the name of the page to retrieve and the action to execute. If this service is unable to execute, an error message is displayed to the user.
- ❖ This service executes the sub-service LOAD_DOC_ENVIRONMENT. This sub-service loads the template page configuration information for the content item environment.

Required Parameters

These parameters must be specified:

Parameter	Description
Action	The action to execute. This is usually <i>GetTemplatePage</i> .
IdcService	Must be set to GET_DOC_PAGE.
Page	The HTML page requested.

Example

- ❖ To get the home page the parameters will be:


```
IdcService=GET_DOC_PAGE
Page=HOME_PAGE
Action=GetTemplatePage
```
- ❖ To get the standard query page the parameters will be:

```
IdcService=GET_DOC_PAGE  
Page= STANDARD_QUERY_PAGE  
Action=GetTemplatePage
```

GET_DOC_SUBSCRIBERS

Description

This service displays the subscriber list.

- ❖ Given a content item name, the service returns the list of subscribers.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to get subscriber list for "{dDocName}".

Required Parameters

These parameters must be specified:

Parameter	Description
dDocName	The content item identifier (Content ID).
IdcService	Must be set to GET_DOC_SUBSCRIBERS.



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Example

```
IdcService=GET_DOC_SUBSCRIBERS
```

```
dDocName=test_000036
```

GET_DOCEXTENSIONS

Description

This service returns a list of content item extensions defined in the system and the format each is mapped to.

- ❖ Returns the properties information and *ExtensionFormatMap* result set. It does not need any other parameters.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve extensions.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_DOCEXTENSIONS.

Example

```
IdcService=GET_DOCEXTENSIONS
```

GET_DOCFORMATS

Description

This service returns a list of content item formats and their associated conversion methods and descriptions.

- ❖ Returns the properties information and *DocFormats* result set. It does not need any other parameters.
- ❖ See also GET_DOC_CONFIG_INFO.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve file formats.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_DOCFORMATS.

Example

```
IdcService=GET_DOCFORMATS
```

GET_DOCTYPES

Description

This service returns a list of the content item types, their descriptions, and the associated gif.

- ❖ Returns the properties information and *DocType* result set. It does not need any other parameters.
- ❖ See also GET_DOC_CONFIG_INFO.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve file formats.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_DOCTYPES.

Example

```
IdcService=GET_DOCTYPES
```

GET_DOCUMENT_PROBLEMREPORTS

Description

This content publisher related service provides a content item problem report.

- ❖ Given a content item name and a project ID, the service returns the problem report for that content item. Problem report information is generated through the workflow process.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve content item "{dDocName}" problem reports.

Required Parameters

These parameters must be specified:

Parameter	Description
dDocName	The content item identifier (Content ID).
dProjectID	The unique project ID generated by content server.
IdcService	Must be set to GET_DOCUMENT_PROBLEMREPORTS.



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Example

```
IdcService=GET_DOCUMENT_PROBLEMREPORTS
dProjectID=servG0SGXDW5YD55T80NHPR57MWW40
dDocName=security_000015
```

GET_DYNAMIC_CONVERSION

Description

This service returns the latest revision, or the latest released revision of a content item.

- ❖ Given a *dID* or a *dDocName* along with a *RevisionSelectionMethod* parameter, the service determines the filename of a particular rendition of the revision and returns that file to the client.
- ❖ The most likely errors are some form of mismatched parameters or a request for a content item that does not exist. If this service is unable to execute, an error message is displayed to the user.
- ❖ The *GET_DYNAMIC_CONVERSION* service is used by the Dynamic Converter component to request dynamic HTML output by content item revision ID or by content item name and to request the latest revision or the latest released revision of a content item.
- ❖ This service computes the *dID* (content item revision ID) for the revision, and then determines the filename of a particular rendition of the revision with the computed *dID*. A specified *dID* or a *dDocName* (content item name) along with a *RevisionSelectionMethod* parameter can be used.



Note: It is recommended that *dDocName* be present in all requests for content items where the requester knows the *dDocName*. Error messages in the content server assume that it is present, as do other features such as forms.

Required Parameters



Important: Either the content item revision ID (*dID*) must be specified or a content item name (*dDocName*) along with a *RevisionSelectionMethod*

parameter must be defined.

Parameter	Description
IdcService	Must be set to GET_DYNAMIC_CONVERSION.

Optional Parameters

These optional parameters may be specified:

Parameter	Description
conversionTemplate	<p>The selected conversion template.</p> <p>The argument <i>conversionTemplate</i> can be passed to the GET_DYNAMIC_CONVERSION service to directly set the template. This overrides the template conversions criteria page.</p>
DCViewFormat	<p>The viewing format.</p> <p>The argument <i>DCViewFormat</i> can be passed to the GET_DYNAMIC_CONVERSION service to specify directly which file is to be converted. The values are Native or WebViewable.</p> <ul style="list-style-type: none"> ❖ If the value is <i>Native</i>, the content item is viewed in the native format for that file. ❖ If the value is <i>WebViewable</i>, the content item is viewed in a web viewable format.

Parameter	Description
dDocName	<p>The content item identifier (Content ID).</p> <ul style="list-style-type: none"> ❖ If dDocName is not present, dID must be present and RevisionSelectionMethod must not be present. ❖ If RevisionSelectionMethod is present, a rendition of a revision of the content item with this name will be returned, if it exists. If RevisionSelectionMethod is not present, dDocName may be used in error messages.
dID	<p>The generated content item revision ID.</p> <ul style="list-style-type: none"> ❖ If dID is not specified, dDocName and RevisionSelectionMethod must be specified. ❖ A rendition of the revision of the content item with this ID will be returned, if it exists, and the RevisionSelectionMethod parameter does not exist or has the value <i>Specific</i>.
RevisionSelection Method	<p>The revision selection method.</p> <ul style="list-style-type: none"> ❖ If present, dDocName must be present. The value of this variable is the method used to compute a dID from the specified dDocName. Its value may be <i>Specific</i>, <i>Latest</i>, or <i>LatestReleased</i>. ❖ If the value is <i>Specific</i>, the dDocName is ignored, and dID is required and is used to get a rendition. If the value is <i>Latest</i>, the latest revision of the content item is used to compute the dID. If the value is <i>LatestReleased</i>, the latest released revision of the content item is used to compute the dID.



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific

rendition of a content item.

Example

❖ IdcCommand command file format:

```
# Retrieve a web-viewable revision
IdcService=GET_DYNAMIC_CONVERSION
dDocName=corporatereport
RevisionSelectionMethod=LatestReleased
DCViewFormat=WebViewable
```

❖ Serialized HDA format:

```
@Properties LocalData
IdcService=GET_DYNAMIC_CONVERSION
dID=54321
@end
```

GET_DYNAMIC_PAGE

Description

This service retrieves a page for dynamic assembly.

- ❖ This service requires the name of the page to retrieve and the action to execute. Normally, the *PageName* is the name of the page that is created on WebLayout Editor
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve page.

Required Parameters

These parameters must be specified:

Parameter	Description
Action	The action to execute. This is usually <i>GetTemplatePage</i> .
IdcService	Must be set to GET_DYNAMIC_PAGE.
Page	The HTML page requested.

Example

To get the index page the required parameters will be:

```
IdcService=GET_DYNAMIC_PAGE
PageName=index
Action=GetTemplatePage
```

GET_DYNAMIC_URL

Description

This service retrieves a dynamic page.

- ❖ Given a relative URL, the service retrieves a dynamic page.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve file.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_DYNAMIC_URL.

Example

```
IdcService=GET_DYNAMIC_URL
```

GET_EXTERNAL_DOC_INFO

Description

This service retrieve external content item information.

- ❖ Given a content item name, the service retrieves external content item information.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve external content item info for "{dDocName}".

Required Parameters

These parameters must be specified:

Parameter	Description
dDocName	The content item identifier (Content ID).
IdcService	Must be set to GET_EXTERNAL_DOC_INFO.



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Optional Parameters

This parameters may be specified:

Parameter	Description
sCollectionID	The collection ID used by content server to locate the collection.

IdcCommand Services

Example

```
IdcService=GET_EXTERNAL_DOC_INFO
```

```
dDocName=adminform113
```

GET_EXTERNAL_HIGHLIGHT_INFO

Description

This service retrieves external content highlight information.

- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve external content highlight information.

Required Parameters

These parameters must be specified:

Parameter	Description
dDocName	The content item identifier (Content ID).
HighlightType	The highlight type: <ul style="list-style-type: none"> ❖ For PDF, use PdfHighlight. ❖ For HTML, use HtmlHighlight.
IdcService	Must be set to GET_EXTERNAL_HIGHLIGHT_INFO.
QueryText	The user supplied text submitted for the content item query.
SortField	The name of the metadata field to sort on. <ul style="list-style-type: none"> ❖ Examples: dInDate, dOutDate, alternateFile. ❖ Defaults to DOCID.
SortOrder	The sort order. Allowed values are ASC (ascending) and DES (descending).



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Example

```
IdcService=GET_EXTERNAL_HIGHLIGHT_INFO
```

```
dDocName=test113
```

```
HighlightType=PdfHighlight
```

```
QueryText=test
```

```
SortField=dInDate
```

```
SortOrder=Desc
```

GET_EXTERNAL_XML_HIGHLIGHT_INFO

Description

This service retrieves external XML highlight information.

- ❖ This service is called if a user is doing a full-text search for a PDF document in a browser environment. When the user selects the link for the PDF on the result page and views the highlighted word on the PDF.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve external content highlight information.

Required Parameters

These parameters must be specified:

Parameter	Description
dDocName	The content item identifier (Content ID).
HighlightType	The highlight type: <ul style="list-style-type: none"> ❖ For PDF, use PdfHighlight. ❖ For HTML, use HtmlHighlight.
IdcService	Must be set to GET_EXTERNAL_XML_HIGHLIGHT_INFO
QueryText	The user supplied text submitted for the content item query.
SortField	The name of the metadata field to sort on. <ul style="list-style-type: none"> ❖ Examples: dInDate, dOutDate, alternateFile. ❖ Defaults to DOCID.
SortOrder	The sort order. Allowed values are ASC (ascending) and DES (descending).



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Example

```
IdcService=GET_EXTERNAL_XML_HIGHLIGHT_INFO  
dDocName=test113  
HighlightType=PdfHighlight  
QueryText=test  
SortField=dInDate  
SortOrder=Desc
```

GET_FILE

Description

This service returns a specific rendition of a content item, the latest revision, or the latest released revision. A copy of the file is retrieved without performing a check out.

- ❖ This command computes the dID (content item revision ID) for the revision, and then determines the filename of a particular rendition of the revision with the computed dID. A specified dID or a dDocName (content item name) along with a RevisionSelectionMethod parameter can be used.
- ❖ Given a dID or a dDocName along with a RevisionSelectionMethod parameter, the service determines the filename of a particular rendition of the revision and returns that file to the client. This service is used from a browser and returns a response page (compare GET_FILE_BY_NAME).
- ❖ The most likely errors are some form of mismatched parameters or a request for a revision or rendition that does not exist. If this service is unable to execute, an error message is displayed to the user.



Note: It is recommended that *dDocName* be present in all requests for content items where the requester knows the *dDocName*. Error messages in the content server assume that it is present, as do other features such as forms.

Required Parameters



Important: Either the content item revision ID (*dID*) must be specified or a content item name (*dDocName*) along with a *RevisionSelectionMethod*

parameter must be defined.

Parameter	Description
IdcService	Must be set to GET_FILE.

Optional Parameters

These optional parameters may be specified:

Parameter	Description
dDocName	<p>The content item identifier (Content ID).</p> <ul style="list-style-type: none"> ❖ If dDocName is not present, dID must be present and RevisionSelectionMethod must not be present. ❖ If RevisionSelectionMethod is present, a rendition of a revision of the content item with this name will be returned, if it exists. If RevisionSelectionMethod is not present, dDocName may be used in error messages.
dID	<p>The generated content item revision ID.</p> <ul style="list-style-type: none"> ❖ If dID is not specified, dDocName and RevisionSelectionMethod must specified. ❖ A rendition of the revision of the content item with this ID will be returned, if it exists, and the RevisionSelectionMethod parameter does not exist or has the value <i>Specific</i>.

Parameter	Description
Rendition	<p>The content item rendition. This parameter specifies the rendition of the content item and can be set to <i>Primary</i>, <i>Web</i>, or <i>Alternate</i>. If Rendition is not present, it defaults to <i>Primary</i>.</p> <ul style="list-style-type: none"> ❖ If the value is <i>Primary</i>, the primary rendition of the selected revision is returned. ❖ If the value is <i>Web</i>, the web viewable rendition of the selected revision is returned. ❖ If the value is <i>Alternate</i>, the alternate rendition of the selected revision is returned.
RevisionSelection Method	<p>The revision selection method.</p> <ul style="list-style-type: none"> ❖ If present, dDocName must be present. The value of this variable is the method used to compute a dID from the specified dDocName. Its value may be <i>Specific</i>, <i>Latest</i>, or <i>LatestReleased</i>. ❖ If the value is <i>Specific</i>, the dDocName is ignored, and dID is required and is used to get a rendition. If the value is <i>Latest</i>, the latest revision of the content item is used to compute the dID. If the value is <i>LatestReleased</i>, the latest released revision of the content item is used to compute the dID.



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Example

- ❖ IdcCommand command file format:

```
# Retrieve a web-viewable revision
```

IdcCommand Services

```
IdcService=GET_FILE  
dDocName=corporatereport  
RevisionSelectionMethod=LatestReleased  
Rendition=Web
```

❖ Serialized HDA format:

```
@Properties LocalData  
IdcService=GET_FILE  
dID=54321  
@end
```

GET_FILE_BY_NAME

Description

This service returns a specific rendition of a content item, the latest revision, or the latest released revision. A copy of the file is retrieved without performing a check out.

- ❖ This command computes the *dID* (content item revision ID) for the revision, and then determines the filename of a particular rendition of the revision with the computed *dID*. A specified *dID* or a *dDocName* (content item name) along with a *RevisionSelectionMethod* parameter can be used.
- ❖ Given a *dID* or a *dDocName* along with a *RevisionSelectionMethod* parameter, the service determines the filename of a particular rendition of the revision and returns that file to the client. This service is used by an applet or application (compare *GET_FILE*).
- ❖ The most likely errors are some form of mismatched parameters or a request for a revision or rendition that does not exist. If this service is unable to execute, an error message is displayed to the user.



Note: It is recommended that *dDocName* be present in all requests for content items where the requester knows the *dDocName*. Error messages in the content server assume that it is present, as do other features such as forms.

Required Parameters



Important: Either the content item revision ID (*dID*) must be specified or a content item name (*dDocName*) along with a *RevisionSelectionMethod*

parameter must be defined.

Parameter	Description
IdcService	Must be set to GET_FILE_BY_NAME.

Optional Parameters

These optional parameters may be specified:

Parameter	Description
dDocName	<p>The content item identifier (Content ID).</p> <ul style="list-style-type: none"> ❖ If dDocName is not present, dID must be present and RevisionSelectionMethod must not be present. ❖ If RevisionSelectionMethod is present, a rendition of a revision of the content item with this name will be returned, if it exists. If RevisionSelectionMethod is not present, dDocName may be used in error messages.
dID	<p>The generated content item revision ID.</p> <ul style="list-style-type: none"> ❖ If dID is not specified, dDocName and RevisionSelectionMethod must be specified. ❖ A rendition of the revision of the content item with this ID will be returned, if it exists, and the RevisionSelectionMethod parameter does not exist or has the value <i>Specific</i>.

Parameter	Description
Rendition	<p>The content item rendition. This parameter specifies the rendition of the content item and can be set to <i>Primary</i>, <i>Web</i>, or <i>Alternate</i>. If Rendition is not present, it defaults to <i>Primary</i>.</p> <ul style="list-style-type: none"> ❖ If the value is <i>Primary</i>, the primary rendition of the selected revision is returned. ❖ If the value is <i>Web</i>, the web viewable rendition of the selected revision is returned. ❖ If the value is <i>Alternate</i>, the alternate rendition of the selected revision is returned.
RevisionSelection Method	<p>The revision selection method.</p> <ul style="list-style-type: none"> ❖ If present, dDocName must be present. The value of this variable is the method used to compute a dID from the specified dDocName. Its value may be <i>Specific</i>, <i>Latest</i>, or <i>LatestReleased</i>. ❖ If the value is <i>Specific</i>, the dDocName is ignored, and dID is required and is used to get a rendition. If the value is <i>Latest</i>, the latest revision of the content item is used to compute the dID. If the value is <i>LatestReleased</i>, the latest released revision of the content item is used to compute the dID.



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Example

```
IdcService=GET_GET_FILE_BY_NAME
dDocName=notice
```

IdcCommand Services

```
RevisionSelectionMethod=LatestReleased  
Rendition=web
```

GET_FILELIST

Description

This service returns the list of files from the application directory.

- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve file list.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_FILELIST.

Example

```
IdcService=GET_FILELIST
```

GET_HIGHLIGHT_INFO

Description

This service retrieves highlight information.

- ❖ Given a content item name, the service retrieves highlight information for that content item.
- ❖ This service is run from the browser interface when you do a full-text search and click the Content ID or thumbnail on the search result page. On the displayed page the words that you searched for are highlighted. For HTML document the words are bold and for PDF document they are highlighted.
- ❖ The most likely errors are when the content item no longer exists in the system or if the user fails the security check. The service throws reasonable exceptions for display to the user in these situations.

Required Parameters

These parameters must be specified:

Parameter	Description
dDocName	The content item identifier (Content ID).
IdcService	Must be set to GET_HIGHLIGHT_INFO.
QueryText	The user supplied text submitted for the content item query.
SortField	The name of the metadata field to sort on. <ul style="list-style-type: none"> ❖ Examples: dInDate, dOutDate, alternateFile. ❖ Defaults to DOCID.
SortOrder	The sort order. Allowed values are ASC (ascending) and DES (descending).



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Optional Parameters

These parameters may be specified:

Parameter	Description
dWebExtension	The file extension of the web viewable content. For example, HTML, PDF, TXT.
HighlightType	The highlight type: <ul style="list-style-type: none"> ❖ For PDF, use PdfHighlight. ❖ For HTML, use HtmlHighlight.

Example

- ❖ IdcCommand command file format:

```
IdcService=GET_HIGHLIGHT_INFO
dDocName=test_000043
QueryText=service
SortField=dInDate
SortOrder=Desc
```

- ❖ Serialized HDA format with optional parameters:

```
@Properties LocalData
IdcService=GET_HIGHLIGHT_INFO
dDocName=test_000043
QueryText=service
SortField=dInDate
```

IdcCommand Services

```
SortOrder=Desc  
HighlightType=HtmlHighlight  
dWebExtension=html  
@end
```

GET_METADEFs

Description

This service returns a list of the metadata definition fields and their attributes.

- ❖ This service returns all the custom metadata information that you add in the Information Field on the Configuration Manager applet.
- ❖ Returns the properties information and *MetaFieldInfo* result set. It does not need any other parameters.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to get meta definition configuration.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_METADEFs.

Example

```
IdcService=GET_METADEFs
```

GET_OPTION_LIST

Description

This service provides a table of all option lists.

- ❖ Each list has a common key value, option value, and order.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve option list.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_OPTION_LIST.

Example

```
IdcService=GET_OPTION_LIST
```

GET_PORTAL_PAGE

Description

This service displays the portal page.

- ❖ Used for editing personalization. This service requires the name of the page to retrieve and the action to execute.
- ❖ The service loads configuration information for the template page, retrieves the option lists for custom fields, and dynamically displays the portal page.
- ❖ The most likely errors are when there is an error loading configuration information, an error retrieving the options list, or when the referenced HTML page is not found. The service throws reasonable exceptions for display to the user in these situations.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_PORTAL_PAGE.

Example

- ❖ To get the portal design page the parameters will be:


```
IdcService=GET_PORTAL_PAGE
Action=GetTemplatePage
Page=PNE_PORTAL_DESIGN_PAGE
```
- ❖ To get to edit the saved queries the parameters will be:


```
IdcService=GET_PORTAL_PAGE
```

IdcCommand Services

```
Action=GetTemplatePage  
Page=PNE_PORTAL_SAVED_QUERIES_PAGE
```

GET_PROBLEMREPORTS_SEARCH_FORM

Description

This workflow related service retrieves the problem report search form.

- ❖ This service does not require any extra parameters. This service is related to staging workflow in Content Publisher.
- ❖ Problem report information is generated through the workflow process.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve problem reports search form.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_PROBLEMREPORTS_SEARCH_FORM.

Example

```
IdcService=GET_PROBLEMREPORTS_SEARCH_FORM
```

GET_PROBLEMREPORTS_SEARCH_RESULTS

Description

This workflow related service provides a table that contains problem report search result information.

- ❖ This service does not require any extra parameters to retrieve all the problem reports (like doing blank query search). You can use additional parameters to get more specific information. This is related to staging workflow.
- ❖ Problem report information is generated through the workflow process.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve problem reports.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_PROBLEMREPORTS_SEARCH_RESULTS.

Optional Parameters

These parameters may be specified:

Parameter	Description
dDocTitle	The content item title.

Parameter	Description
dPrSeverity	The problem report severity level such as CRITICAL, MODERATE, MINOR. This option has no bearing on how a problem report is handled within the workflow process.
dPrAuthor	The problem report author.
dPrCaption	The caption for the problem report.
dPrState	The problem report state such as OPEN, CLOSED, FIXED.
dDocName	The content item identifier (Content ID).
dPrjDescription	The project description.



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Example

```
IdcService=GET_PROBLEMREPORTS_SEARCH_RESULT
```

GET_PROJECT_INFO

Description

This workflow related service provides a table that stores information about all content items associated with a project.

- ❖ Given a project ID, the service retrieves a table that stores information about all content items associated with a Content Publisher project.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve project registration information for "{dProjectID}".

Required Parameters

These parameters must be specified:

Parameter	Description
dProjectID	The unique project ID generated by content server.
IdcService	Must be set to GET_PROJECT_INFO.

Example

IdcService=GET_PROJECT_INFO

dProjectID=servG0SGXDW5YD55T80NHPR57MWW40

GET_PROJECT_REGISTRATION

Description

This workflow related service retrieves project registration information.

- ❖ Given a project ID, the service retrieves the associated project registration information.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve project registration information for "{dProjectID}".

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_PROJECT_REGISTRATION.
projectID	The unique project ID generated by content server.



Note: Use *projectID* for this service (not *dProjectID*).

Example

```
IdcService=GET_PROJECT_REGISTRATION
projectID=saviZN0LN18G9D6THX91KD409RTMKG
```

GET_PROVIDER_INFO

Description

This service retrieves provider information.

- ❖ Given a provider name, the service retrieves the defined provider information. This parameter is sufficient if you are executing from IdcCommand. If you are running this service from the browser environment, you must pass the *ResourceTemplate* parameter.
- ❖ If this service is unable to execute, this message is displayed to the user: *Unable to retrieve provider info for "{pName}"*.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_PROVIDER_INFO.
pName	The provider name.

Additional Parameters

This parameter may be required:

Parameter	Description
ResourceTemplate	The resource template. Required when running the service from a browser interface. For example: ResourceTemplate=DATABASE_PROVIDER_INFO

Example

```
IdcService=GET_PROVIDER_INFO  
pName=SystemDatabase
```

GET_PROXYED_ARCHIVECOLLECTIONS

Description

The service returns the list of archive collections on a proxied server.

- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve the list of collections.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_PROXYED_ARCHIVECOLLECTIONS.
psIDC_Name	The proxied content server instance.

Example

```
IdcService=GET_PROXYED_ARCHIVECOLLECTIONS
```

```
psIDC_Name=Proxied_2_on_test13
```

GET_REGISTEREDPROJECTS

Description

This workflow related service provides a table that stores information about any registered projects.

- ❖ The service returns a table that stores information about any projects registered through Content Publisher.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve registered projects.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_REGISTEREDPROJECTS.

Example

```
IdcService=GET_REGISTEREDPROJECTS
```

GET_REPLICATION_DATA

Description

This service retrieves replication data.

- ❖ Given the collection name, this service returns replication information.
- ❖ The IDC_Name is normally the name of the collection. If you are using the default collection then this is the name of the content server instance.
- ❖ The most likely error is an incorrect archive location. If this service is unable to execute, this message is displayed to the user: *Unable to retrieve the replication data.*



Caution: Using duplicate IDC_Name collection names will cause data corruption. The Archiver cannot be used to move or copy data between two instances that share the same IDC_Name. To do so will corrupt the data on the target system.

Required Parameters

These parameters must be specified:

Parameter	Description
IDC_Name	The name of the content server instance.
IdcService	Must be set to GET_REPLICATION_DATA.

Example

```
IdcService=GET_REPLICATION_DATA
```

```
IDC_Name=Master_on_server01
```

GET_RESULT_OPTIONS

Description

This service returns the information about search result page/search result template.

- ❖ Not currently used. This service retrieves shared table information from the Verity templates.
- ❖ This is a legacy service used for reverse compatibility and is not called directly by any current code.

Example

This service is referenced in the standard services HTM file:

```
<install_dir>\shared\config\std_services.htm
```

GET_SEARCH_RESULTS

Description

This service retrieves the search results for the passed query text.

- ❖ Used to display the search results to a user making a content item query.
- ❖ You can append values for Title, Content ID, etc., on the *QueryText* parameter to refine this service.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve search results.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_SEARCH_RESULTS.
QueryText	The user supplied text submitted for the content item query.

Optional Parameters

These parameters may be specified:

Parameter	Description
resultCount	The number of results to return, defaults to "25"
sortField	The name of the metadata field to sort on. ❖ Examples: dInDate, dDocTitle, Score. ❖ Defaults to dInDate.
sortOrder	The sort order. Allowed values are ASC (ascending) and DES (descending).
startRow	The row to begin the search results. For example, if a result returns 200 rows, and resultCount is 25, set startRow to 26 to obtain the second set of results.

Example

```
IdcService=GET_SEARCH_RESULTS
```

```
QueryText=benefits
```

GET_SECURE_PAGE

Description

This service executes a secure page request.

- ❖ This service requires the name of the page to retrieve and the action to execute.
- ❖ The most likely error is when the referenced HTM page is not found. If this service is unable to execute, an error message is displayed to the user.
- ❖ This service executes the sub-service LOAD_DOC_ENVIRONMENT. This sub-service loads the template page configuration information for the content item environment.

Required Parameters

These parameters must be specified:

Parameter	Description
Action	The action to execute. This is usually <i>GetTemplatePage</i> .
IdcService	Must be set to GET_SECURE_PAGE.
Page	The HTML page requested.

Example

To get the content management page the parameters will be:

```
IdcService=GET_SECURE_PAGE
```

```
Page=DOC_MANAGEMENT_LINKS
```

```
Action=GetTemplatePage
```

GET_SELF_REGISTER_PAGE

Description

This service retrieves the self registration page.

- ❖ Used for editing personalization.
- ❖ The service retrieves the self registration page and loads the user meta data information for user self registration.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_SELF_REGISTER_PAGE.

Example

```
IdcService=GET_SELF_REGISTER_PAGE
```

GET_SERVER_OUTPUT

Description

This Admin Server related service retrieves the server output.

- ❖ Used to load the server output for display.
- ❖ No additional parameters are required when executing this service through IdcCommand. If you run the service from the browser environment, you must include the *IDC_Name* parameter.
- ❖ The *IDC_Name* is normally the name of the collection. If you are using the default collection then this is the name of the content server instance.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to output from server.



Caution: Using duplicate *IDC_Name* collection names will cause data corruption. The Archiver cannot be used to move or copy data between two instances that share the same *IDC_Name*. To do so will corrupt the data on the target system.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_SERVER_OUTPUT.

Additional Parameters

This parameters may be required:

Parameter	Description
IDC_Name	Required if you run the service from the browser environment. The name of the content server instance.

Example

```
IdcService=GET_SERVER_OUTPUT
```

GET_TABLE

Description

This service exports the specified table in the content server database.

- ❖ Exports the specified table by creating a result set and adding it to the serialized *hda*. If the table is not found the service will fail. It is up to the calling program receiving the serialized *hda* to store this result set for later usage.
- ❖ The most likely error is a table name that does not exist. If this service is unable to execute, an error message is displayed to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_TABLE.
tableName	The name of table to export.

Example

- ❖ IdcCommand command file format:

```
IdcService=GET_TABLE
tableName=Users
```

- ❖ Serialized HDA format:

```
@Properties LocalData
IdcService=GET_TABLE
tableName=Users
@end
```

GET_UPDATE_FORM

Description

This form return related service builds the update form.

- ❖ Given a content revision ID, the service builds the associated update form. The named content item must be an active form and must be updated with the Submit control on the web viewable rendition.
- ❖ The most likely error is naming a content item that is no longer in the system. If this service is unable to execute, this message is displayed to the user: *Unable to build update form for "{dDocName}"*.

Required Parameters

These parameters must be specified:

Parameter	Description
dID	The generated content item revision ID.
IdcService	Must be set to GET_UPDATE_FORM.

Example

```
IdcService=GET_UPDATE_FORM
```

```
dID=59
```

GET_UPDATE_PROBLEMREPORT_FORM

Description

This workflow related service retrieves the problem report update form.

- ❖ Given a problem report ID, the service retrieves the problem report update form from the database.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve problem report update form.

Required Parameters

These parameters must be specified:

Parameter	Description
dPrID	The problem report ID.
IdcService	Must be set to GET_UPDATE_PROBLEMREPORT_FORM.

Example

```
IdcService=GET_UPDATE_PROBLEMREPORT_FORM  
dPrID=1
```

GET_USER_INFO

Description

This service retrieves the user list.

- ❖ Given a defined user, the service retrieves the user list.
- ❖ If this service is run from the browser environment, the content server returns the user information for the currently logged in user.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve user list.

Required Parameters

These parameters must be specified:

Parameter	Description
dUser	The defined user.
IdcService	Must be set to GET_USER_INFO.

Example

```
IdcService=GET_USER_INFO
```

```
dUser=sysadmin
```

GET_USERS

Description

This service retrieves a list of all users.

- ❖ Returns a list of all users registered in the system with their primary attributes: username, full name, password, email address, directory, old style role (when only one role was given each user), type, and password encoding.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve user list.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_USERS.

Example

```
IdcService=GET_USERS
```

GET_WF_TEMPLATE

Description

This workflow related service retrieves workflow template information.

- ❖ Given a workflow template name, this service returns the template description.
- ❖ The most likely error is a workflow template name that does not exist. If this service is unable to execute, this message is displayed to the user:
Unable to retrieve data for template "{dWfTemplateName}".
- ❖ Compare GET_WF_TEMPLATES.

Required Parameters

These parameters must be specified:

Parameter	Description
dWfTemplateName	The workflow template name.
IdcService	Must be set to GET_WF_TEMPLATE.

Example

```
IdcService=GET_WF_TEMPLATE
dWfTemplateName=servicestemplate
```

GET_WF_TEMPLATES

Description

This workflow related service returns all workflow template names and their descriptions.

- ❖ The most likely error is a workflow name that does not exist. If this service is unable to execute, this message is displayed to the user: *Unable to retrieve workflow templates.*
- ❖ Compare GET_WF_TEMPLATE.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_WF_TEMPLATES.

Example

IdcService=GET_WF_TEMPLATES

GET_WORKFLOW

Description

This workflow related service retrieves workflow information.

- ❖ Given a workflow name, the service retrieves workflow information.
- ❖ The most likely error is a workflow name that does not exist. If this service is unable to execute, this message is displayed to the user: *Unable to retrieve information for the workflow "{dWfName}"*.

Required Parameters

These parameters must be specified:

Parameter	Description
dWfName	The workflow name.
IdcService	Must be set to GET_WORKFLOW.

Example

```
IdcService=GET_WORKFLOW
```

```
dWfName=D20
```

GET_WORKFLOW_INFO

Description

This workflow related service retrieves the Workflow Step information for content (usually in active workflow).

- ❖ Given a content item ID and a workflow contribution stage, the service retrieves content item workflow information.
- ❖ This service is executed in a browser interface by selecting Active Workflows from PNE links, clicking on the workflow name, and then clicking the link for step name.
- ❖ The dWfStepID parameter is required for this service. The content server automatically assigns dWfStepID for that contribution stage.
- ❖ The most likely error is a content item or workflow name that does not exist. The service throws reasonable exceptions for display to the user in these situations.

Required Parameters

These parameters must be specified:

Parameter	Description
dID	The generated content item revision ID.
dWfStepID	The workflow contribution stage ID. This auto-generated value can be retrieved from the database table.
IdcService	Must be set to GET_WORKFLOW_INFO.

Example

```
IdcService=GET_WORKFLOW_INFO
```

```
dID=61
```

```
dWfStepID=19
```

GET_WORKFLOWDOCREVISIONS

Description

This workflow related service retrieves the content item revision list for a workflow.

- ❖ Given a workflow name, the service retrieves the content item list of revisions for that workflow.
- ❖ This service is run in the browser environment by selecting Active Workflows in PNE links and clicking on the workflow name.
- ❖ The most likely error is a workflow name that does not exist. If this service is unable to execute, this message is displayed to the user: *Unable to retrieve content item list for workflow "{dWfName}"*.

Required Parameters

These parameters must be specified:

Parameter	Description
dWfName	The workflow name.
IdcService	Must be set to GET_WORKFLOWDOCREVISIONS.

Example

```
IdcService=GET_WORKFLOW
```

```
dWfName=D20
```

GET_WORKFLOWDOCUMENTS

Description

This workflow related service retrieves the content item list for a workflow.

- ❖ Given a workflow name, the service retrieves the content item list for that workflow. The service provides a list of all content items in workflows and is updated by the system server to keep track of the status of content items (state and step) that are in workflows.
- ❖ The most likely error is a workflow name that does not exist. If this service is unable to execute, this message is displayed to the user: *Unable to retrieve content item list for workflow "{dWfName}"*.

Required Parameters

These parameters must be specified:

Parameter	Description
dWfName	The workflow name.
IdcService	Must be set to GET_WORKFLOWDOCUMENTS.

Example

```
IdcService=GET_WORKFLOWDOCUMENTS
```

```
dWfName=c2
```

GET_WORKFLOWS

Description

This workflow related service retrieves a list of all workflows.

- ❖ Retrieves a list of all workflows including their description, security group, status, and type.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve workflow list.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to GET_WORKFLOWS.

Example

IdcService=GET_WORKFLOWS

GET_XML_HIGHLIGHT_INFO

Description

This service retrieves XML highlight information.

- ❖ This service is called if a user is doing a full-text search for a PDF document in a browser environment. When the user selects the link for the PDF on the result page and views the highlighted word on the PDF.
- ❖ The most likely error is a content item name that does not exist. The service throws reasonable exceptions for display to the user in these situations.

Required Parameters

These parameters must be specified:

Parameter	Description
dDocName	The content item identifier (Content ID).
HighlightType	The highlight type: <ul style="list-style-type: none"> ❖ For PDF, use PdfHighlight. ❖ For HTML, use HtmlHighlight.
IdcService	Must be set to GET_XML_HIGHLIGHT_INFO.
QueryText	The user supplied text submitted for the content item query.
SortField	The name of the metadata field to sort on. <ul style="list-style-type: none"> ❖ Examples: dInDate, dOutDate, alternateFile. ❖ Defaults to DOCID.
SortOrder	The sort order. Allowed values are ASC (ascending) and DES (descending).



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Example

```
IdcService=GET_XML_HIGHLIGHT_INFO  
dDocName=test13  
QueryText=service  
SortField=dInDate  
SortOrder=Desc  
dDocName=TEST13  
HighlightType=PdfHighlight
```

IMPORT_ARCHIVE

Description

This service imports a given archive.

- ❖ Given a collection and an archive in the collection, this service will attempt to import the archive. The import will read in the archive's definition file and will follow the import rules specified. For this service to successfully execute, the IdcCommand script must specify if documents, users and/or document configuration information is to be imported.
- ❖ The server executes this service asynchronously. This means that the server needs to be running for the services to be even executed and that IdcCommand will exit announcing success, when actually all it has done is told the server to perform the action.
- ❖ The server can only export or import archives one at a time. A batch file that has multiple exports and imports will declare success on the first and failure on all subsequent commands until it has finished the first. This means that a batch file should have only one export or import in it and some external process will need to determine that the action has completed before issuing another export/import command.
- ❖ The most likely error is a archive name that does not exist. If this service is unable to execute, this message is displayed to the user: *Unable to import archive "{aArchiveName}"*.
- ❖ The IDC_Name is normally the name of the collection. If you are using the default collection then this is the name of the content server instance.



Caution: Using duplicate IDC_Name collection names will cause data corruption. The Archiver cannot be used to move or copy data between two instances that share the same IDC_Name. To do so will corrupt the data on the target system.

Required Parameters

These parameters must be specified:

Parameter	Description
aArchiveName	The archive name.
IDC_Name	The collection name or content server instance the collection belongs to.
IdcService	Must be set to IMPORT_ARCHIVE.



Note: One of the listed *Additional Parameters* must be set for this service to successfully execute.

Additional Parameters

One of these parameters must be set:

Parameter	Description
aImportDocuments	When set to true, the service will import the previously exported revisions.
aImportDocConfig	When set to true, the service will import the previously exported document configuration information. This currently consists only of doc type information.  Important: If the <i>document configuration</i> has not been exported and this flag is set, the service will fail.

Parameter	Description
aImportUsers	<p>When set to true, the service will import the previously exported users.</p> <p> Important: If the <i>users</i> have not been exported and this flag is set, the service will fail</p>

Example

- ❖ IdcCommand command file format (import the revisions in the archive “archive_test” from the collection “Master_on_server01”):

```
# To import an archive
IdcService=IMPORT_ARCHIVE
aArchiveName=archive_test
IDC_Name=Master_on_server01
aImportDocuments=true
```

- ❖ Serialized HDA format:

```
@Properties LocalData
IdcService=IMPORT_ARCHIVE
aArchiveName=archive_test
IDC_Name=Master_on_server01
aImportDocuments=true
@end
```

IMPORT_DOCUMENT

Description

This service imports a content item.

- ❖ Given a content item name, the service imports the content item.
- ❖ The IDC_Name is normally the name of the collection. If you are using the default collection then this is the name of the content server instance.
- ❖ The most likely error is a content item name that does not exist. If this service is unable to execute, this message is displayed to the user: *Unable to import content item "{dDocName}"*.



Caution: Using duplicate IDC_Name collection names will cause data corruption. The Archiver cannot be used to move or copy data between two instances that share the same IDC_Name. To do so will corrupt the data on the target system.

Required Parameters

These parameters must be specified:

Parameter	Description
aArchiveName	The archive name.
dDocName	The content item identifier (Content ID).
IDC_Name	The name of the content server instance.
IdcService	Must be set to IMPORT_DOCUMENT.



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Example

```
IdcService=IMPORT_DOCUMENT  
IDC_Name=Master_on_server01  
aArchiveName=JAN_22_02  
dDocName=billing_00004
```

LOAD_GLOBALINCLUDES

Description

This service returns the portal page content.

- ❖ Used during the page assembly process for dynamic pages.
- ❖ Using Web Layout Editor applet, select Options—Update Portal. The content shown on that screen is the content that is returned from content server if you run this service.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to load global includes.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to LOAD_GLOBALINCLUDES.

Example

```
IdcService=LOAD_GLOBALINCLUDES
```

LOAD_RESOURCE_FILE

Description

This service returns the specified resource file.

- ❖ The LOAD_RESOURCE_FILE command computes the dID for the revision, and then determines the filename of a particular rendition of the revision with the computed dID. A specified dID or a dDocName along with a RevisionSelectionMethod parameter can be used.
- ❖ Given a dID or a dDocName along with a RevisionSelectionMethod parameter, the service determines the filename of a particular rendition of the revision and returns that file to the client
- ❖ The most likely errors are some form of mismatched parameters or a request for a revision or rendition that does not exist. The service throws reasonable exceptions for display to the user in these situations.



Note: It is recommended that *dDocName* be present in all requests for content items where the requester knows the *dDocName*. Error messages in the content server assume that it is present, as do other features such as forms.

Required Parameters



Important: Either the content item revision ID (*dID*) must be specified or a content item name (*dDocName*) along with a *RevisionSelectionMethod* parameter must be defined.

Parameter	Description
IdcService	Must be set to LOAD_RESOURCE_FILE.

Optional Parameters

These optional parameters may be specified:

Parameter	Description
dDocName	<p>The content item identifier (Content ID).</p> <ul style="list-style-type: none"> ❖ If dDocName is not present, dID must be present and RevisionSelectionMethod must not be present. ❖ If RevisionSelectionMethod is present, a rendition of a revision of the content item with this name will be returned, if it exists. If RevisionSelectionMethod is not present, dDocName may be used in error messages.
dID	<p>The generated content item revision ID.</p> <ul style="list-style-type: none"> ❖ If dID is not specified, dDocName and RevisionSelectionMethod must be specified. ❖ A rendition of the revision of the content item with this ID will be returned, if it exists, and the RevisionSelectionMethod parameter does not exist or has the value <i>Specific</i>.
Rendition	<p>The content item rendition. This parameter specifies the rendition of the content item and can be set to <i>Primary</i>, <i>Web</i>, or <i>Alternate</i>. If Rendition is not present, it defaults to <i>Primary</i>.</p> <ul style="list-style-type: none"> ❖ If the value is <i>Primary</i>, the primary rendition of the selected revision is returned. ❖ If the value is <i>Web</i>, the web viewable rendition of the selected revision is returned. ❖ If the value is <i>Alternate</i>, the alternate rendition of the selected revision is returned.

Parameter	Description
RevisionSelectionMethod	<p>The revision selection method.</p> <ul style="list-style-type: none"> ❖ If present, dDocName must be present. The value of this variable is the method used to compute a dID from the specified dDocName. Its value may be <i>Specific</i>, <i>Latest</i>, or <i>LatestReleased</i>. ❖ If the value is <i>Specific</i>, the dDocName is ignored, and dID is required and is used to get a rendition. If the value is <i>Latest</i>, the latest revision of the content item is used to compute the dID. If the value is <i>LatestReleased</i>, the latest released revision of the content item is used to compute the dID.



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Example

```
IdcService=LOAD_RESOURCE_FILE
dID=456
```

LOGIN

Description

This page request service forces a log in and executes an HTML page request.

- ❖ This service is used to login to the content server. If you are logging in from the browser the parameter Auth is also required.
- ❖ This service requires the name of the page to retrieve and the action to execute. If this service is unable to execute, an error message is displayed to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
Action	The action to execute. Usually set to <i>GetTemplatePage</i> .
IdcService	Must be set to LOGIN
Page	The HTML page requested.

Additional Parameters

This parameters may be required:

Parameter	Description
Auth	Required when logging in from a browser interface. For example: Auth=Internet

Example

Displays the home page when you login to the content server:

```
IdcService=LOGIN
```

```
Page=HOME_PAGE
```

```
Action=GetTemplatePage
```

MERGE_TABLE

Description

This service allows the user to add rows to any table in the content server database.

- ❖ Given a table name in the system and a result set with the same name, the service will attempt to insert or update the rows in the specified table.
- ❖ If no constraints are defined, the service will attempt to insert the row.
- ❖ If constraints are defined, the service will first do a lookup and if the designated row is found, an update is performed. Otherwise, the row will be inserted.
- ❖ All errors are logged to the content server log. If this service is unable to execute, an error message is displayed to the user.



Important: By default, this service is not error tolerant and does not execute the merge in a transaction. Consequently, if the `isErrorTolerant` or `isTransactional` flags are not set to `TRUE` and if the service encounters an error, it will abort where the error occurred and not rollback the already updated and inserted rows. In this situation, if you do not specify constraints and rerun the merge table, the merge will fail in Microsoft SQL Server and ORACLE due to constraint violation.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to MERGE_TABLE.
tableName	The name of the table to merge.



Important: The ResultSet definition for the table containing the rows to insert or update must also be specified.

Optional Parameters

These optional parameters may be specified:

Parameter	Description
constraintKeys	Specifies a comma-separated list of column names used as unique identifiers. The service will initially do a look up with these constraints. If the row is found, the service does an update. Otherwise, a new row is inserted.
isDeleteTable	If TRUE, the service deletes the table before attempting the merge. The delete is performed in a transaction with the merge, if <code>isTransactional</code> is TRUE. By default, this value is FALSE.
isErrorTolerant	If TRUE, will not abort when errors are encountered during the merge. By default, this value is FALSE.
isTransactional	If TRUE, the merge is performed in a transaction. If an error occurs during the merge and <code>isErrorTolerant</code> is FALSE, the merge will abort and rollback all the changes. By default, this value is FALSE.

Example

❖ IdcCommand command file format:

```
# Merge new entries into the Alias table
IdcService=MERGE_TABLE
tableName=Alias
constraintKeys=dAlias
```

IdcCommand Services

```
# Rows to be updated or inserted
@ResultSet Alias
2
dAlias
dAliasDescription
MyAlias
Just a test alias
@end
<<EOD>>
```

```
# Delete the alias table and insert new rows, do a
rollback if something fails.
IdcService=MERGE_TABLE
tableName=Alias
isTransactional=true
isDeleteTable=true
```

```
# Rows to be updated or inserted
@ResultSet Alias
2
dAlias
dAliasDescription
MyAlias
Just a test alias
@end
<<EOD>>
```

❖ Serialized HDA format (Example 1):

```
@Properties LocalData
IdcService=MERGE_TABLE
tableName=Alias
constraintKeys=dAlias
@end
@ResultSet Alias
2
dAlias
dAliasDescription
MyAlias
Just a test alias
@end
```

❖ Serialized HDA format defining a table name and result set and inserting a new row into the database (inserts new row into Users table):

```
@Properties LocalData
IdcService=MERGE_TABLE
tableName=Users
@end
@ResultSet Users
2
dName
dUserAuthType
user15
LOCAL
@end
```

NOTIFY_CHANGE

Description

This service notifies the provider of changes.

- ❖ Given a content server instance the service notifies the provider of changes to the content item.
- ❖ The IDC_Name is normally the name of the collection. If you are using the default collection then this is the name of the content server instance.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to notify "{InstanceMenuLabel}" of change.



Caution: Using duplicate IDC_Name collection names will cause data corruption. The Archiver cannot be used to move or copy data between two instances that share the same IDC_Name. To do so will corrupt the data on the target system.

Required Parameters

These parameters must be specified:

Parameter	Description
IDC_Name	The name of the content server instance.
IdcService	Must be set to NOTIFY_CHANGE.

Example

```
IdcService=NOTIFY_CHANGE
```

```
IDC_Name=Master_on_server01
```

NOTIFY_CONTRIBUTOR

Description

This workflow related service notifies a contributor.

- ❖ Given a content revision ID, the service notifies the contributor on the problem report for the specified content item.
- ❖ The most likely error is a content item no longer in the system. If this service is unable to execute, this message is displayed to the user: *Unable to notify on problem report for content item "{dDocName}"*

Required Parameters

These parameters must be specified:

Parameter	Description
dID	The generated content item revision ID.
IdcService	Must be set to NOTIFY_CONTRIBUTOR.

Example

```
IdcService=NOTIFY_CONTRIBUTOR.
```

```
dID=55
```

PAGE_HANDLER

Description

This service rebuilds the static web layout structure as it is defined in the Web Layout Editor.

- ❖ Returns the entire page list in the serialized *hda*.
- ❖ This service is asynchronous and the action is performed in a background thread. This service can only be executed in the context of the server and only completes successfully if a connection is made to the server.
- ❖ If this service is unable to execute, this message is displayed to the user:
Failed page service action.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to PAGE_HANDLER.
IsRebuild	Enables a rebuild. <ul style="list-style-type: none"> ❖ Must be set to TRUE (1) for the server to rebuild the web layout pages. ❖ Default is FALSE (0).
PageFunction	Must be set to GetPageList.

Example

- ❖ IdcCommand command file format (rebuild the web layout pages):

```
# Rebuild the entire search index
```

```
IdcService=PAGE_HANDLER  
PageFunction=GetPageList  
IsRebuild=1
```

❖ Serialized HDA format (rebuild the web layout pages):

```
@Properties LocalData  
IdcService=PAGE_HANDLER  
PageFunction=GetPageList  
IsRebuild=1  
@end
```

PING_SERVER

Description

This service evaluates whether a connection to the server exists.

- ❖ Returns status information.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to establish connection to the server.



Tech Tip: Execute a PING_SERVER request before calling other services to ensure that there is a connection to the content server and that you are logged in as a user authorized to execute commands.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to PING_SERVER.

Example

Serialized HDA format (see Figure 4-1 for sample return information):

```
@Properties LocalData
IdcService=PING_SERVER
IsJava=1
Auth=Internet
@end
```

```
Content-type: text/html
Content-Length: 421

<?hda version="6.0 SNAP-020207" jcharset=Cp1252
encoding=iso-8859-1?>

@Properties LocalData
dUser=sysadmin
blFieldTypes=StatusMessage message
refreshSubMonikers=
StatusMessage=You are logged in as 'sysadmin'.
loadedUserAttributes=1
blDateFormat=M/d{/yy} {h:mm[:ss]
{aa}[zzz]}!tAmerica/Chicago!mAM,PM
changedSubjects=
refreshSubjects=
Auth=Internet
refreshMonikers=
changedMonikers=
IdcService=PING_SERVER
IsJava=1
@end
```

Figure 5-1 PING_SERVER sample return information.

PNE_GET_SEARCH_RESULTS

Description

This service returns the search results to the personal navigation environment for display to the user.

- ❖ Service used for editing personalization.
- ❖ This service is run if you perform a search on the portal page.
- ❖ You can append values for Title, Content ID, etc., on the *QueryText* parameter to refine this service.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve search results.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to PNE_GET_SEARCH_RESULTS.
QueryText	The user supplied text submitted for the content item query.

Example

```
IdcService=PNE_GET_SEARCH_RESULTS
```

```
QueryText=benefits
```

PNE_SAVE_QUERY

Description

This service saves the query results to the personal navigation environment for display to the user.

- ❖ Used for editing personalization.
- ❖ This service requires the name of the page to retrieve and the action to execute.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to open the portal page.

Required Parameters

These parameters must be specified:

Parameter	Description
Action	The action to execute. Usually set to <i>GetTemplatePage</i> .
IdcService	Must be set to PNE_SAVE_QUERY.
Page	The HTML page requested.

Example

```
IdcService=PNE_SAVE_QUERY  
Page=HOME_PAGE  
Action=GetTemplatePage
```

PNE_UPDATE_PERSONAL_URLS

Description

This service updates the personal URLs of the personal navigation environment for display to the user.

- ❖ Used for editing personalization.
- ❖ This service is execute from a browser interface to update the personal URL links. This service requires the name of the page to retrieve and the action to execute.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to open the portal design page.

Required Parameters

These parameters must be specified:

Parameter	Description
Action	The action to execute. Usually set to <i>GetTemplatePage</i> .
IdcService	Must be set to PNE_UPDATE_PERSONAL_URLS.
Page	The HTML page requested.
titleEd	The “title” or link name provided by the user on the browser interface.
websiteEd	The URL selected by the user.



Note: The specific link name and URL information such as *titleEd* and *websiteEd* is provided by the user and passed as variables to update the PNE_PORTAL_PERSONAL_URLS_PAGE page.

Example

```
IdcService=PNE_UPDATE_PERSONAL_URLS  
Page=PNE_PORTAL_PERSONAL_URLS_PAGE  
Action=GetTemplatePage  
titleEd=Stellent  
websiteEd=http://www.stellent.com
```

PNE_UPDATE_PORTAL_INFO

Description

This service updates the portal information of the personal navigation environment for display to the user.

- ❖ Used for editing personalization.
- ❖ This service updates the portal information. For example to show or hide a link to Portal Design page on PNE links. This service requires the name of the page to retrieve and the action to execute.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to open the portal design page.

Required Parameters

These parameters must be specified:

Parameter	Description
Action	The action to execute. Usually set to <i>GetTemplatePage</i> .
IdcService	Must be set to PNE_UPDATE_PORTAL_INFO.
Page	The HTML page requested.

Example

```
IdcService=PNE_UPDATE_PORTAL_INFO
Page=PNE_PORTAL_DESIGN_PAGE
Action=GetTemplatePage
```

PROBLEMREPORT_INFO

Description

This workflow related service returns problem report information.

- ❖ Given a problem report ID, the service retrieves the related problem report information.
- ❖ The most likely error is a misnamed problem report caption. If this service is unable to execute, this message is displayed to the user: *Unable to retrieve info on problem report.*

Required Parameters

These parameters must be specified:

Parameter	Description
dPrID	The problem report ID.
IdcService	Must be set to PROBLEMREPORT_INFO.

Example

```
IdcService=PROBLEMREPORT_INFO
```

```
dPrID=1
```

QUERY_DOC_ACCOUNTS

Description

This service returns all the accounts in the content server.

- ❖ No additional parameters are required.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to get accounts.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to QUERY_DOC_ACCOUNTS.

Example

IdcService=QUERY_DOC_ACCOUNTS

QUERY_GROUP

Description

This service returns the specified group name and its description.

- ❖ Given a group name, the service returns the group information.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to edit group "{dGroupName}".

Required Parameters

These parameters must be specified:

Parameter	Description
dGroupName	The defined group name.
IdcService	Must be set to QUERY_GROUP.

Example

```
IdcService=QUERY_GROUP  
dGroupName=Public
```

QUERY_USER_ATTRIBUTES

Description

This service returns user attributes.

- ❖ Given a user name, the service returns the defined user attributes.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to get attributes for user "{dName}".

Required Parameters

These parameters must be specified:

Parameter	Description
dName	The unique name.
IdcService	Must be set to QUERY_USER_ATTRIBUTES.

Example

```
IdcService=QUERY_USER_ATTRIBUTES
```

```
dName=jsmith
```

REGISTER_IMPORTER

Description

This service registers the importer for the archive.

- ❖ The *IsRegister* parameter must be set with a Boolean value and the *EditItems* parameter must be set to *aRegisteredImporter,aImportLogonUser*.
- ❖ The *IDC_Name* is normally the name of the collection. If you are using the default collection then this is the name of the content server instance.
- ❖ The most likely errors are mismatched parameters or an incorrect instance menu label or archive name. If this service is unable to execute, this message is displayed to the user: *Unable to (un)register importer for the archive "{aArchiveName}" of "{InstanceMenuLabel}"*.



Caution: Using duplicate *IDC_Name* collection names will cause data corruption. The Archiver cannot be used to move or copy data between two instances that share the same *IDC_Name*. To do so will corrupt the data on the target system.

Required Parameters

These parameters must be specified:

Parameter	Description
<i>aArchiveName</i>	The archive name.
<i>EditItems</i>	Must be set to <i>aRegisteredImporter,aImportLogonUser</i>
<i>IDC_Name</i>	The name of the content server instance.
<i>IdcService</i>	Must be set to REGISTER_IMPORTER .

Parameter	Description
IsRegister	Defines registration setting. <ul style="list-style-type: none"> ❖ Set to TRUE (1) to Register Self. ❖ Set to FALSE (0) to Unregister.

Example

Self register the content server as importer:

```

IdcService=REGISTER_IMPORTER
IDC_Name=Master_on_server01
aArchiveName=JAN_22_02
IsRegister=1
EditItems=aRegisteredImporter,aImportLogonUser
    
```

REGISTER_USER

Description

This service registers the user.

- ❖ Used for editing personalization.
- ❖ Given a user name, the service registers the user.
- ❖ Only the listed Required Parameters are needed to execute this command. However, the user will be a local user with guest role and blank password by default if you don't pass any additional parameters.
- ❖ The most likely error is a user name that is not unique. The service throws reasonable exceptions for display to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
dName	The unique name.
IdcService	Must be set to REGISTER_USER.



Note: If you do not pass any optional parameters, the user will be a local user with guest role and the password will be blank by default.

Optional Parameters

These parameters may be specified:

Parameter	Description
dEmail	The email address for the user.

Parameter	Description
dUserLocale	The locale designation such as English-US, English-UK, Deutsch, Français, Español.
dUserType	The defined user type.
dFullName	The full name of user.
dPassword	The password for the user.

Example

IdcService=REGISTER_USER

dName=user20

REMOVE_COLLECTION

Description

This service removes a collection.

- ❖ Given a collection name, the service removes the collection.
- ❖ You cannot remove the default collection. Any additional added collections can be removed using this service.
- ❖ The IDC_Name is normally the name of the collection. If you are using the default collection then this is the name of the content server instance.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to remove the collection "{InstanceMenuLabel}".



Caution: Using duplicate IDC_Name collection names will cause data corruption. The Archiver cannot be used to move or copy data between two instances that share the same IDC_Name. To do so will corrupt the data on the target system.

Required Parameters

These parameters must be specified:

Parameter	Description
IDC_Name	The name of the content server instance.
IdcService	Must be set to REMOVE_COLLECTION.

Example

```
IdcService=REMOVE_COLLECTION
```

```
IDC_Name=new_collection
```

IdcCommand Services

REMOVE_EXPORTER

Description

This service removes an exporter.

- ❖ The IDC_Name is normally the name of the collection. If you are using the default collection then this is the name of the content server instance.
- ❖ The most likely error is an incorrect archive location. If this service is unable to execute, this message is displayed to the user: *Unable to remove exporter "{aArchiveLocation}"*.



Caution: Using duplicate IDC_Name collection names will cause data corruption. The Archiver cannot be used to move or copy data between two instances that share the same IDC_Name. To do so will corrupt the data on the target system.

Required Parameters

These parameters must be specified:

Parameter	Description
aArchiveName	The archive name.
IDC_Name	The name of the content server instance.
IdcService	Must be set to REMOVE_EXPORTER.

Example

```
IdcService=REMOVE_EXPORTER
IDC_Name=Master_on_server02
aArchiveName=archive_02
```

REMOVE_IMPORTER

Description

This service removes an importer.

- ❖ The IDC_Name is normally the name of the collection. If you are using the default collection then this is the name of the content server instance.
- ❖ The most likely error is an incorrect archive location. If this service is unable to execute, this message is displayed to the user: *Unable to remove importer "{aArchiveLocation}"*.



Caution: Using duplicate IDC_Name collection names will cause data corruption. The Archiver cannot be used to move or copy data between two instances that share the same IDC_Name. To do so will corrupt the data on the target system.

Required Parameters

These parameters must be specified:

Parameter	Description
aArchiveName	The archive name.
IDC_Name	The name of the content server instance.
IdcService	Must be set to REMOVE_EXPORTER.

Example

```
IdcService=REMOVE_EXPORTER
IDC_Name=Master_on_server01
aArchiveName=my_archive_test
```

REQUEST_SECURITYINFO

Description

This service retrieves all security and user information from the content server.

- ❖ Provider related service.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve security info for "{InstanceMenuLabel}".

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to REQUEST_SECURITYINFO.

Example

```
IdcService=REQUEST_SECURITYINF
```

RESEND_PROBLEMREPORT

Description

This is a staging workflow related service that notifies a contributor.

- ❖ Given a problem report ID, the service sends email to the contributor.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to resend problem report '{dPrCaption}'.

Required Parameters

These parameters must be specified:

Parameter	Description
dPrID	The problem report ID.
IdcService	Must be set to RESEND_PROBLEMREPORT.

Example

```
IdcService=RESEND_PROBLEMREPORT
```

```
dPrID=1
```

RESUBMIT_FOR_CONVERSION

Description

This service resubmits the content item if it failed the initial conversion attempt.

- ❖ Inbound Refinery related service. Given a content item name, the service resubmits the content item for conversion.
- ❖ The most likely errors are when the content item no longer exists in the system or when the user does not have the security level to perform this action. If this service is unable to execute, an error message is displayed to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
dID	The generated content item revision ID.
IdcService	Must be set to RESUBMIT_FOR_CONVERSION.

Example

```
IdcService=RESUBMIT_FOR_CONVERSION
```

```
dID=62
```

REV_HISTORY

Description

This service retrieves the revision history.

- ❖ Given a content item revision ID, the service retrieves the revision history for that content item.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve the revision history.

Required Parameters

These parameters must be specified:

Parameter	Description
dID	The generated content item revision ID.
IdcService	Must be set to REV_HISTORY.

Example

IdcService=REV_HISTORY

dID=62

SAVE_GLOBALINCLUDES

Description

This service loads saves all global includes used during the page assembly process for dynamic pages.

- ❖ This service is run when you Update the portal page from Web Layout Editor. Set the *IsRebuild* parameter to TRUE to update the portal page.
- ❖ If this service is unable to execute, this message is displayed to the user: *Unable to save global includes.*

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to SAVE_GLOBALINCLUDES.
IsRebuild	Enables a rebuild. <ul style="list-style-type: none"> ❖ Must be set to TRUE (1) for the server to update the portal page. ❖ Default is FALSE (0).
PageFunction	Must be set to GetPageList.

Example

```
IdcService=SAVE_GLOBALINCLUDES
PageFunction=GetPageList
IsRebuild=1
```

SELECTDOC

Description

This service selects the named content item.

- ❖ Used only by ODMA.
- ❖ Given a content item revision ID, the service retrieves content item information and evaluates the user security level.
- ❖ If this service is unable to execute, or if the user does not have permission to access the content item, this message is displayed to the user: *Unable to select file "{dDocName}"*.

Required Parameters

These parameters must be specified:

Parameter	Description
dID	The generated content item revision ID.
IdcService	Must be set to SELECTDOC.

Example

```
IdcService=SELECTDOC
```

```
dID=62
```

START_SEARCH_INDEX

Description

This service either updates or rebuilds the search index.

- ❖ This service is useful for starting an automated rebuild of the index.
- ❖ The command rebuilds the entire search index if `IsRebuild` is set to `TRUE`. Otherwise, it performs an update. This service is asynchronous and the action is performed in a background thread. This service can only be executed in the context of the server and only completes successfully if a connection is made to the server.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to start search indexing session.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to <code>START_SEARCH_INDEX</code> .

Optional Parameters

These optional parameters may be specified:

Parameter	Description
IsRebuild	<p>Enables a rebuild.</p> <ul style="list-style-type: none"> ❖ Set to TRUE (1), for the server to do a complete rebuild of the search index ❖ Set to FALSE (0) for the server to do an update only. ❖ Default is FALSE.

Example

- ❖ IdcCommand command file format:

```
# Rebuild the entire search index
IdcService=START_SEARCH_INDEX
IsRebuild=true
```

- ❖ Serialized HDA format (rebuild the search index).

```
@Properties LocalData
IdcService=START_SEARCH_INDEX
IsRebuild=true
@end
```

SUBMIT_HTML_FORM

Description

This service submits an HTML form.

- ❖ Given a defined HTML form (dID), the service submits the HTML form.
- ❖ Value of any fields in the form can be submitted as additional parameters.
- ❖ This service is run to submit HTML forms. The pre-condition is that an HTML form exists on the system and the value dID is that of the HTML form. Usually the Auto Content ID is turned on to submit HTML forms, if not then each submitted form must be assigned a unique content ID.
- ❖ The most likely error is a defined HTML form that does not exist. If this service is unable to execute, this message is displayed to the user: *Unable to begin form submission.*

Required Parameters

These parameters must be specified:

Parameter	Description
dID	The generated content item revision ID. For this service, the dID must be an HTML form.
IdcService	Must be set to SUBMIT_HTML_FORM.

Optional Parameters

You can submit any text field in the form as an optional parameter and pass a value to that field. For example, if the form has a text field called *product*, the string value *publisher* could be assigned to that field (see example below).

Example

- ❖ IdcCommand command file format:

```
IdcService=SUBMIT_HTML_FORM  
dID=44
```

- ❖ Serialized HDA format with an optional parameter (assigns a value to a text field):

```
@Properties LocalData  
IdcService=SUBMIT_HTML_FORM  
dID=44  
Product=publisher  
@end
```

SUBSCRIBE

Description

This service adds a subscription.

- ❖ This service is used from a browser and returns a response page (compare SUBSCRIBE_EX).
- ❖ The most likely errors are when the content item no longer exists in the system or when the user does not have the security level to perform this action. If this service is unable to execute, an error message is displayed to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
dID	The generated content item revision ID.
dSubscriptionEmail	The email address for the subscription.
dSubscriptionType	The subscription type.
IdcService	Must be set to SUBSCRIBE.

Example

These parameters are passed in the browser environment to execute this command:

```
<input type=hidden name=IdcService value="SUBSCRIBE">
<input type=hidden name=dID value="63">
<input type=hidden name=dSubscriptionType
value=my_subscription>
```

IdcCommand Services

```
<input type=hidden name=dSubscriptionEmail  
value=sysadmin@stellent.com>
```

SUBSCRIBE_DOC_USER

Description

This service is used in the browser environment when a user subscribes to a content item (builds the subscription page).

- ❖ This service is executed when a user subscribes to a document (performs a search and clicks subscribe). If the system has more than one subscription type the user will be redirected to the Subscription page where the user can subscribe to different criteria.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to build subscription page for "{dDocName}".

Required Parameters

These parameters must be specified:

Parameter	Description
dDocName	The content item identifier (Content ID).
dID	The generated content item revision ID.
IdcService	Must be set to SUBSCRIBE_DOC_USER.



Note: Additional parameters are passed when the user subscribes to the content item (see example below).

Example

These parameters are passed in the browser environment to execute this command:

```
<form name=SubscriptionForm action="/intradoc-cgi/idc_cgi_isapi.dll"
```

IdcCommand Services

```
method="GET" ">
<input type=hidden name=dID value="63">
<input type=hidden name=dDocName value="test_000045">
<input type=hidden name=IdcService
value="SUBSCRIBE_DOC_USER">
<input type=hidden name=subscribeService value=SUBSCRIBE>
<input type=hidden name=exitUrl value="/intradoc-
cgi/idc_cgi_isapi.dll?IdcService=DOC_INFO&dID=63&dDocName=t
est_000045">
<input type=hidden name=title value="Subscriptions">
<input type=hidden name=unsubscribeService
value=UNSUBSCRIBE>
<input type=submit value="Subscriptions">
</form>
```

SUBSCRIBE_EX

Description

This service adds a user alias for a criteria subscription.

- ❖ This service is used by an applet or application (compare SUBSCRIBE).
- ❖ The most likely errors are mismatched parameters, a misnamed subscription alias type or a misnamed a subscription alias. If this service is unable to execute, this message is displayed to the user: *Unable to subscribe {dSubscriptionAliasType} "{dSubscriptionAlias}"*

Required Parameters

These parameters must be specified:

Parameter	Description
dSubscriptionAlias	The user alias for the criteria subscription.
dSubscriptionAlias Type	The alias subscription type.
dSubscriptionEmail	The email address for the subscription.
dSubscriptionID	The unique subscription ID.
dSubscriptionType	The subscription type.
IdcService	Must be set to SUBSCRIBE_EX.

Additional Parameters

For return information about who is subscribed these parameters must be specified:

Parameter	Description
dataSource	The data source.
resultName	The result set name.
whereClause	The programmatic clause.

Example

```
IdcService=SUBSCRIBE_EX  
dSubscriptionID=user20  
dSubscriptionAlias=sysadmin  
dSubscriptionAliasType=user  
dSubscriptionType=test  
dSubscriptionEmail=user20@stellent.com  
resultName=USER_LIST  
dataSource=Subscriptions  
whereClause=dSubscriptionType='test'
```

SUBSCRIPTION_LIST

Description

This service retrieves the subscription list for the logged in user.

- ❖ Used in a browser environment.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve subscriptions for "{dUser}".

Required Parameters

These parameters must be specified:

Parameter	Description
dUser	The unique name.
IdcService	Must be set to SUBSCRIPTION_LIST

Example

```
IdcService=SUBSCRIPTION_LIST
```

```
dUser=sysadmin
```

TEST_PROVIDER

Description

This service tests the provider.

- ❖ Provider related service.
- ❖ Given a provider name, the service tests the defined provider.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to TEST_PROVIDER.
pName	The provider name.

Example

```
IdcService=TEST_PROVIDER
```

```
pName=provider22
```

UNDO_CHECKOUT

Description

This service reverses a content item checkout.

- ❖ Given a content item revision ID, this service attempts to locate the content item in the system and undo the checkout.
- ❖ 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. This service is used from a browser and returns a response page (compare UNDO_CHECKOUT_BY_NAME).
- ❖ The most likely error is a content item name that does not exist. If this service is unable to execute, an error message is displayed to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
dID	The generated content item revision ID.
IdcService	Must be set to UNDO_CHECKOUT.

Example

- ❖ IdcCommand command file format:

```
IdcService=UNDO_CHECKOUT
dID=44
```

- ❖ Serialized HDA format:

```
@Properties LocalData
```

IdcCommand Services

```
IdcService=UNDO_CHECKOUT  
dID=44  
@end
```

UNDO_CHECKOUT_BY_NAME

Description

This service reverses a content item checkout using the Content ID.

- ❖ Given a content item name, this service attempts to locate the content item in the system and undo the checkout.
- ❖ 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. This service is used by an applet or application (compare UNDO_CHECKOUT).
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to undo checkout for "{dDocName}".

Required Parameters

These parameters must be specified:

Parameter	Description
dDocName	The content item identifier (Content ID).
IdcService	Must be set to UNDO_CHECKOUT_BY_NAME.



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Optional Parameters

This optional parameter may be specified:

Parameter	Description
dDocTitle	The content item title.

Example

- ❖ IdcCommand command file format:

```
IdcService=UNDO_CHECKOUT_BY_NAME
dDocName=myDocument
```

- ❖ Serialized HDA format with optional parameter:

```
@Properties LocalData
IdcService=UNDO_CHECKOUT_BY_NAME
dDocName=myDocument
dDocTitle=Just a title
@end
```

UNSUBSCRIBE

Description

This service unsubscribes a content item.

- ❖ This service is executed from a browser interface. This services assumes that a basic subscription is enabled and a content item is subscribed (click Subscribe from the interface). From the Subscription page select the subscription info for that content. You will be redirected to the Document List page. Click Unsubscribe to execute this command.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to unsubscribe "{dDocName}".

Required Parameters

These parameters must be specified:

Parameter	Description
dID	The generated content item revision ID.
dSubscriptionID	The unique subscription ID.
dSubscriptionType	The subscription type.
IdcService	Must be set to UNSUBSCRIBE.

Example

```
IdcService=UNSUBSCRIBE
dID=66
dSubscriptionType=test
dSubscriptionID=test_000048
```

UNSUBSCRIBE_FROM_LIST

Description

This service unsubscribes the user.

- ❖ This service is executed from a browser interface and unsubscribes the currently logged on user from the subscription. This service is run when you go to Subscription page, and click the Unsubscribe link if you have any subscription.
- ❖ If this service is unable to execute, this message is displayed to the user: *Unable to retrieve subscriptions for "{dUser}"*.

Required Parameters

These parameters must be specified:

Parameter	Description
dSubscriptionID	The unique subscription ID.
dSubscriptionType	The subscription type.
IdcService	Must be set to UNSUBSCRIBE_FROM_LIST.

Example

```
IdcService=UNSUBSCRIBE_FROM_LIST
dSubscriptionType=test
dSubscriptionID=test_000046
```

UNSUBSCRIBE_FROM_LIST_EX

Description

This service unsubscribes a user from a specified subscription type.

- ❖ This service is normally run on the Repository Manager applet.
- ❖ This service assumes that you have added a criteria subscription and defined a subscriber (user or alias) to an author. If you delete that subscriber, this service is executed.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to get list of users for subscription type "{dSubscriptionType}".

Required Parameters

These parameters must be specified:

Parameter	Description
dSubscriptionAlias	The user alias for the criteria subscription.
dSubscriptionAlias Type	The alias subscription type.
dSubscriptionEmail	The email address for the subscription.
dSubscriptionID	The unique subscription ID.
dSubscriptionType	The subscription type.
IdcService	Must be set to SUBSCRIBE_EX.

Additional Parameters

For return information about who is subscribed these parameters must be specified:

Parameter	Description
dataSource	The data source.
resultName	The result set name.
whereClause	The programmatic clause.

Example

Unsubscribes *user20* from any content that is checked in by *sysadmin*:

```
IdcService=UNSUBSCRIBE_FROM_LIST_EX
dSubscriptionID=sysadmin
dSubscriptionAlias=user20
dSubscriptionAliasType=user
dSubscriptionType=test
dSubscriptionEmail=sysadmin@stellent.com
resultName=USER_LIST
dataSource=Subscriptions
whereClause=dSubscriptionType='test'
```

UPDATE_DOCINFO

Description

This service updates content item information.

- ❖ This service is used by an applet or application (compare UPDATE_DOCINFO_BYFORM).
- ❖ This service is used by the Repository Manager.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to update the content item information for "{dDocName}".
- ❖ This service executes the sub-service UPDATE_DOCINFO_SUB. This sub-service updates content item information.

Required Parameters

These parameters must be specified:

Parameter	Description
dDocName	The content item identifier (Content ID).
dID	The generated content item revision ID.
dRevLabel	The content item revision label.
dSecurityGroup	The security group such as PUBLIC or SECURE.
IdcService	Must be set to UPDATE_DOCINFO.



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Additional Parameters

This parameter may be required:

Parameter	Description
dDocAccount	The security account for the content item. If you have accounts enabled, you must pass this parameter.

Optional Parameters

You can submit any metadata field in the content item as an optional parameter and pass a value to that field. For example, if the content item had a metadata field called *xComments*, the string value *Public Info* could be assigned to that field (see example below).

Example

- ❖ IdcCommand command file format:

```
IdcService=UPDATE_DOCINFO
dID=66
dDocName=test_000048
dSecurityGroup=Secure
dRevLabel=2
```

- ❖ Serialized HDA format with additional and optional parameters (assigns a value to a metadata field):

```
@Properties LocalData
IdcService=UPDATE_DOCINFO
dID=66
dDocName=test_000048
dSecurityGroup=Secure
```

```
dRevLabel=2  
dDocAccount=mainaccount  
xComments=Public Info  
@end
```

UPDATE_DOCINFO_BYFORM

Description

This service updates content item information.

- ❖ This service is used from a browser and returns a response page (compare UPDATE_DOCINFO).
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to update the content item information for "{dDocName}".
- ❖ This service executes the sub-service UPDATE_DOCINFO_SUB. This sub-service updates content item information.

Required Parameters

These parameters must be specified:

Parameter	Description
dDocName	The content item identifier (Content ID).
dID	The generated content item revision ID.
dRevLabel	The content item revision label.
dSecurityGroup	The security group such as PUBLIC or SECURE.
IdcService	Must be set to UPDATE_DOCINFO_BYFORM.



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Additional Parameters

This parameter may be required:

Parameter	Description
dDocAccount	The security account for the content item. If you have accounts enabled, you must pass this parameter.

Optional Parameters

You can submit any metadata field as an optional parameter and pass a value to that field. For example, the metadata field *dDocAuthor*, could be passed and the string value *Jennifer* assigned to that field (see example below).

Example

- ❖ IdcCommand command file format:

```
IdcService=UPDATE_DOCINFO_BYFORM
dID=66
dSecurityGroup=Secure
dDocName=test_000048
dRevLabel=2
```

- ❖ Serialized HDA format with additional and optional parameters (assigns a value to a metadata field):

```
@Properties LocalData
IdcService=UPDATE_DOCINFO_BYFORM
dID=66
dSecurityGroup=Secure
dDocName=test_000048
```

IdcCommand Services

```
dRevLabel=2  
dDocAccount=mainaccount  
dDocAuthor=Jennifer  
@end
```

UPDATE_DOCINFO_WITH_FILE

Description

This service updates content item information for that file.

- ❖ Used in the browser environment. Clicking Update with Check In button on the doc info page redirects you to the Info Update Form where you have the option to update the metadata only or update the metadata with primary file. If check in a primary file along with updated metadata information this service is executed.
- ❖ If doFileCopy is set to TRUE (1), the file will not be deleted from hard drive after successful checkin, otherwise this service will delete the file.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to update the content item information for "{dDocName}".
- ❖ This service executes the sub-service UPDATE_CHECKIN_SEL_SUB. This sub-service updates the check in information for a selected content item.



Important: This service is used if your system is configured to check in metadata only. The config entry for that option is *AllowPrimaryMetaFile=TRUE*.

Required Parameters

These parameters must be specified:

Parameter	Description
dDocName	The content item identifier (Content ID).
dID	The generated content item revision ID.

Parameter	Description
doFileCopy	Set this flag to TRUE (1) or the file will be removed from your hard drive after checkin.
dRevLabel	The content item revision label.
dSecurityGroup	The security group such as PUBLIC or SECURE.
IdcService	Must be set to UPDATE_DOCINFO_WITH_FILE.
primaryFile	<p>The absolute path to the location of the file as seen from the server. Use the forward slash as the file separator.</p> <p>A primary file must be specified unless checking in metadata only. If an alternate file is specified with the primary file, the content refinery will convert the alternate file. Otherwise, the primary file will be converted.</p> <ul style="list-style-type: none"> ❖ If a primary file is not specified, a metafile can be used in its place. Only one metafile can exist though for each content item (i.e. a primary AND alternate meta file cannot co-exist). ❖ If both a primary and alternate file is specified, their extensions must be different.



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Additional Parameters

This parameter may be required:

Parameter	Description
dDocAccount	The security account for the content item. If you have accounts enabled, you must pass this parameter.

Example

```
IdcService=UPDATE_DOCINFO_WITH_FILE
dID=73
dDocName=test_000054
dSecurityGroup=Secure
dDocAccount=mainaccount
primaryFile=c:/test.txt
dRevLabel=2
doFileCopy=1
```

UPDATE_META_TABLE

Description

This service updates the DocMeta table with the newly added or deleted custom metadata columns.

- ❖ This service alters the database adding, altering and deleting columns in the DocMeta table to conform to the DocMetaDefinition table. By default, fields will only be added or changed, not deleted. To delete a field, it must be specifically mentioned in the optional parameter MetaFieldsToDelete.
- ❖ This service is run when you add/delete Information Fields (metadata) in the Configuration Manager applet and Click the Update Database Design button.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to update content item extended fields table design.



Caution: Data may be lost if a column is deleted. This service should be used carefully.

Required Parameters

This parameter must be specified:

Parameter	Description
IdcService	Must be set to UPDATE_META_TABLE.

Optional Parameters

This optional parameter may be specified:

Parameter	Description
MetaFieldsDoDelete	Specifies a comma-separated list of the database columns to delete from the DocMeta table.

Example

- ❖ IdcCommand command file format:

```
# Synchronize the DocMeta table with the design in the
DocMetaDefinition table
IdcService=UPDATE_META_TABLE
```

- ❖ Serialized HDA format (deletes two fields from the database):

```
@Properties LocalData
IdcService=UPDATE_META_TABLE
MetaFieldsToDelete=xInteger,xMemo
@end
```

UPDATE_OPTION_LIST

Description

This service adds or updates the specified option list.

- ❖ Given an option list key, `dKey`, this service updates or adds the option list to the `OptionLists` table, so that the option list reflects the new values.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to update option list.



Note: Old values will be completely replaced by the newly specified values.

Required Parameters

These parameters must be specified:

Parameter	Description
<code>dKey</code>	The option list key. Specifies an option list key to update.
<code>IdcService</code>	Must be set to <code>UPDATE_OPTION_LIST</code> .
<code>OptionListString</code>	The option list string. Specifies a list of options separated by the escape character and new line character (<code>\n</code>).

Example

- ❖ IdcCommand command file format:

```
# Add the new options list 'LocationList' and
# Add values Madrid, Tokyo, London, Washington
IdcService=UPDATE_OPTION_LIST
dKey=LocationList
```

```
OptionListString=Madrid\nTokyo\nLondon\nWashington
```

- ❖ Serialized HDA format (adds the option list LocationList with the values Madrid, Tokyo, London, and Washington):

```
@Properties LocalData  
IdcService=UPDATE_OPTION_LIST  
dKey=LocationList  
OptionListString=Madrid\nTokyo\nLondon\nWashington  
@end
```

UPDATE_PROBLEMREPORT

Description

This workflow related service updates the problem report.

- ❖ Given a problem report ID and a problem report state, the service updates the problem report.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to update problem report '{dPrCaption}'.

Required Parameters

These parameters must be specified:

Parameter	Description
dPrID	The problem report ID.
dPrState	The problem report state such as OPEN, CLOSED, FIXED.
IdcService	Must be set to UPDATE_PROBLEMREPORT.

Optional Parameters

These parameters may be specified:

Parameter	Description
dPrCaption	The caption for the problem report.
dPrSeverity	The problem report severity level such as CRITICAL, MODERATE, MINOR. This option has no bearing on how a problem report is handled within the workflow process.
prMessage	The provided problem report message.

Example

- ❖ IdcCommand command file format:

```
IdcService=UPDATE_PROBLEMREPORT  
dPrID=1  
dPrState=OPEN
```

- ❖ Serialized HDA file format with optional parameters:

```
@Properties LocalData  
IdcService=UPDATE_PROBLEMREPORT  
dPrID=1  
dPrState=OPEN  
prMessage=Description is stored in hda file.  
dPrCaption=change this caption  
dPrSeverity=SERIOUS  
@end
```

UPDATE_RESULT_TEMPLATE

Description

This service updates a result template

- ❖ This service updates a query result template. When you launch the Web Layout Editor applet, select Options, select Query Result Page, and click Add, Delete, or Edit (the template) this service executed.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to update result template "{name}"

Required Parameters

This parameter must be specified:

Parameter	Description
IdcService	Must be set to UPDATE_RESULT_TEMPLATE.



Note: See below for an example of the entries required to add a new query result template in a serialized HDA format.

Example

- ❖ IdcCommand command file format:

```
IdcService=UPDATE_RESULT_TEMPLATE
```

- ❖ Serialized HDA file format (adds a new query result template called my_research_template).

```
@Properties LocalData
```

```
IdcService=UPDATE_RESULT_TEMPLATE
```

```
@end
```

```
@ResultSet ResultPageUpdates
6
name
formtype
filename
outfilename
flexdata
description
my_research_template
ResultsPage
```

```
Text2<${dDocAuthor$}<${dSecurityGroup$}>Text1<${dDocTitle$}>
testing the update template sevice
@end
```



Important: Notice that there are two blank lines between the ResultsPage entry and the Text2 entry. This is required because there are no values for *filename* and *outfilename* (the blank lines define the empty values).

UPDATE_SUBSCRIPTION_NOTIFY

Description

This service updates the current subscription notification field and sends an unsubscribe notification.

- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to update subscription notification field.

Required Parameters

These parameters must be specified:

Parameter	Description
dSubscriptionID	The unique subscription ID.
dSubscriptionType	The subscription type.
IdcService	Must be set to UPDATE_SUBSCRIPTION_NOTIFY.

Example

```
IdcService=UPDATE_SUBSCRIPTION_NOTIFY  
dSubscriptionID=sysadmin  
dSubscriptionType=sysadmin
```

UPDATE_SUBSCRIPTION_TYPE

Description

This service updates the subscription type.

- ❖ Given a subscription type, the service updates the subscription.
- ❖ The most likely error is a misnamed subscription type. If this service is unable to execute, this message is displayed to the user: *Unable to update subscription type "{scpType}"*.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to UPDATE_SUBSCRIPTION_TYPE.
scpType	The unique subscription type (provided name).

Optional Parameters

These parameters may be specified:

Parameter	Description
scpDescription	The subscription description.
scpEnabled	When set to TRUE (1), enables subscriptions based on criteria.
scpFields	The subscription field type (metadata field). For example, dDocAuthor, dDocType.

Example

- ❖ IdcCommand command file format:

```
IdcService=UPDATE_SUBSCRIPTION_TYPE  
scpType=my_subscription
```

- ❖ Serialized HDA file format (changes the subscription for Author and disables the subscription):

```
@Properties LocalData  
IdcService=UPDATE_SUBSCRIPTION_TYPE  
scpType=my_subscription  
scpFields=dDocAuthor  
scpDescription=updating subscription  
scpEnabled=false  
@end
```

UPDATE_SUBSCRIPTION_USED

Description

This service updates the currently used subscription and sends an unsubscribe notification.

- ❖ This service updates the *dSubscriptionUsedDate* column of the Subscription table in the database which is the Access Date. For example, if you have subscribed to the content authored by sysadmin, when content is checked in by sysadmin, you will receive an email. If you click the link to the content on the email message, this service executed.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to update subscription notification field.

Required Parameters

These parameters must be specified:

Parameter	Description
dDocName	The content item identifier (Content ID).
dSubscriptionAlias	The user alias for the criteria subscription.
dSubscriptionAlias Type	The alias subscription type.
dSubscriptionID	The unique subscription ID.
dSubscriptionType	The subscription type.
IdcService	Must be set to UPDATE_SUBSCRIPTION_USED.



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Example

```
IdcService=UPDATE_SUBSCRIPTION_USED  
dSubscriptionType=test_subscription  
dSubscriptionID=sysadmin  
dDocName=test_000056  
dSubscriptionAlias=global  
dSubscriptionAliasType=user
```

UPDATE_USEROPTION_LIST

Description

This service updates the user type list.

- ❖ In the User Admin applet if you add/edit a user you will see a User Type field. If you click the List button and add/delete/modify the Option List for 'User Type' this service executed.
- ❖ This option list information is not stored in the database but in the *useroptions.hda* file which is normally located in the `<stellent_dir>/data/users/config` sub-folder.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to update option list.

Required Parameters

These parameters must be specified:

Parameter	Description
dKey	The option list key. Specifies an option list key to update.
IdcService	Must be set to UPDATE_USEROPTION_LIST
OptionListString	The option list string. Specifies a list of options separated by the escape character and new line character (<code>\n</code>) .

Example

Text:

```
IdcService=
```

Example

- ❖ IdcCommand command file format:

```
# Add values NY, DC, CA to UserTypeList
IdcService=UPDATE_USEROPTION_LIST
dKey=Users_UserTypeList
OptionListString=NY\nDC\nCA
```

- ❖ Serialized HDA format (adds the values NY, DC, and CA to UserTypeList):

```
@Properties LocalData
IdcService=UPDATE_USEROPTION_LIST
dKey=Users_UserTypeList
OptionListString=NY\nDC\nCA
@end
```

UPLOAD_NEW_COMPONENT

Description

This Admin Server related service uploads a new component.

- ❖ Unpackages the content items of the manifest and uploads the new component for installation.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to install the new component

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to UPLOAD_NEW_COMPONENT.

Example

```
IdcService=UPLOAD_NEW_COMPONENT
```

VIEW_DOC

Description

This service retrieve highlight information.

- ❖ Given a content item name, the service evaluates security information and displays the content item highlight information.
- ❖ Usually this service is used when you have external collection rather than content server search collection. But in either case, the search collection must be created by Verity Search engine. For example, if you want to do a search on an external collection, you must provide the VdkVgwKey and sCollectionID parameters so the collection can be located by the content server.
- ❖ Usually, the VdkVgwKey is the dDocName. Verity gives a unique ID to each content, and content server maps that Verity ID (VdkVgwKey) with dDocName (content ID).
- ❖ If this service is unable to execute, or if the user does not have permission to access the content item, this message is displayed to the user: *Unable to retrieve highlight information.*

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to VIEW_DOC.
QueryText	The user supplied text submitted for the content item query.
sCollectionID	The collection ID used by content server to locate the collection.

Parameter	Description
SortField	The name of the metadata field to sort on. ❖ Examples: dInDate, dOutDate, alternateFile. ❖ Defaults to docID.
SortOrder	The sort order. Allowed values are ASC (ascending) and DES (descending).



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Optional Parameters

This parameter may be specified:

Parameter	Description
ViewType	The display format. If this parameter is not defined ViewText is used by default. ❖ Set to ViewHtml to display as HTML. ❖ Set to ViewText to display as Text.

Example

```
IdcService=VIEW_DOC
sCollectionID=external
SortField=dInDate
SortOrder=Desc
QueryText=test
ViewType=ViewText
```

WORK_IN_PROGRESS

Description

This service retrieves the works in progress list.

- ❖ Builds the database result set and returns a list of the content items actively in progress.
- ❖ If this service is unable to execute, this message is displayed to the user:
Unable to retrieve works in progress list.

Required Parameters

These parameters must be specified:

Parameter	Description
IdcService	Must be set to WORK_IN_PROGRESS.

Example

IdcService=WORK_IN_PROGRES

WORKFLOW_APPROVE

Description

This workflow related service approves the content item for the workflow.

- ❖ Given a content item revision ID, the service approves the content item for the workflow.
- ❖ This service approves content at each step in the workflow. For example, you have a criteria workflow with 3 steps. You checkin a content matching that criteria workflow condition. That content will go to the first step of that workflow. When you run this service, it will approve the content on that step and it goes to the second step. If you again run this service, that content will go to the third step.
- ❖ If this service is unable to execute, an error message is displayed to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
dID	The generated content item revision ID.
IdcService	Must be set to WORKFLOW_APPROVE.

Example

```
IdcService=WORKFLOW_APPROVE
```

```
dID=81
```

WORKFLOW_CANCEL

Description

This workflow related service cancels a workflow.

- ❖ Given a workflow name, the service cancels the basic workflow.
- ❖ If this service is unable to execute, an error message is displayed to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
dWfName	The workflow name.
IdcService	Must be set to WORKFLOW_CANCEL.

Example

```
IdcService=WORKFLOW_CANCEL
```

```
dWfName=C10
```

WORKFLOW_CHECKIN

Description

This workflow related service checks a content item into a workflow

- ❖ The *isFinished* attribute is used to indicate the form can no longer be edited (the form allows no more submissions). Used for workflow checkin when editing is finished.
- ❖ If *doFileCopy* is set to TRUE (1), the file will not be deleted from hard drive after successful checkin, otherwise this service will delete the file.
- ❖ The most likely error is a misnamed content item or workflow name. If this service is unable to execute, this message is displayed to the user: *Unable to checkin content item "{dDocName}" for workflow.*
- ❖ This service executes the sub-service WORKFLOW_CHECKIN_SUB. This sub-service checks a content item revision into a workflow.

Required Parameters

These parameters must be specified:

Parameter	Description
dDocName	The content item identifier (Content ID).
dID	The generated content item revision ID.
doFileCopy	Set to TRUE (1) or the file will be removed from your hard drive after checkin.
dRevLabel	The content item revision label.
dSecurityGroup	The security group such as PUBLIC or SECURE.
dWfID	The workflow ID.
dWfName	The workflow name.

Parameter	Description
dWfType	The workflow type such as BASIC or CRITERIA.
IdcService	Must be set to WORKFLOW_CHECKIN.
isFinished	Set to TRUE (1) if this is a workflow check-in and you have finished editing it.
primaryFile	<p>The absolute path to the location of the file as seen from the server. Use the forward slash as the file separator.</p> <p>A primary file must be specified unless checking in metadata only. If an alternate file is specified with the primary file, the content refinery will convert the alternate file. Otherwise, the primary file will be converted.</p> <ul style="list-style-type: none"> ❖ If a primary file is not specified, a metafile can be used in its place. Only one metafile can exist though for each content item (i.e. a primary AND alternate meta file cannot co-exist). ❖ If both a primary and alternate file is specified, their extensions must be different.



Note: Do not confuse the Content ID (dDocName) with the internal content item revision identifier (dID). The *dID* is a generated reference to a specific rendition of a content item.

Additional Parameters

This parameter may be required:

Parameter	Description
dDocAccount	The security account for the content item. If you have accounts enabled, you must pass this parameter.

Example

```
IdcService=WORKFLOW_CHECKIN
dID=92
dDocName=test_00063
dWfID=4
dSecurityGroup=Public
dDocAccount=mainaccount
dWfType=Criteria
dWfName=C10
primaryFile=c:/hello_hello.txt
dRevLabel=2
doFileCopy=1
isFinished=1
```

WORKFLOW_REJECT

Description

This workflow related service rejects a content item from a named workflow.

- ❖ Given a content item name and a workflow name, the service rejects the content item from the workflow.
- ❖ If this service is unable to execute, an error message is displayed to the user.
- ❖ The system administrator cannot approve or reject a step in a workflow unless defined as a reviewer/contributor in that step. Therefore, if you are running this service in IdcCommand as the user *sysadmin*, you cannot reject the step unless *sysadmin* is defined as a reviewer/contributor in that step.

Required Parameters

These parameters must be specified:

Parameter	Description
dID	The generated content item revision ID.
IdcService	Must be set to WORKFLOW_REJECT.
wfRejectMessage	The rejection message.

Example

```
IdcService=WORKFLOW_REJECT
dID=95
wfRejectMessage=Please Revise
```

WORKFLOW_REJECT_FORM

Description

This service rejects a content item from a named workflow.

- ❖ Form return and workflow related service.
- ❖ Given a content item revision ID and a workflow name, the service checks security and rejects the content item from the workflow.
- ❖ If this service is unable to execute, an error message is displayed to the user.
- ❖ This service is executed when a Reviewer/Contributor rejects content in a workflow step (when the reviewer/contributor clicks Reject).
- ❖ This service is normally performed in the browser environment, which uses JavaScript to assign values. Parameters that are passed in the browser environment are listed below.

Required Parameters

These parameters must be specified:

Parameter	Description
dID	The generated content item revision ID.
dWfName	The workflow name.
IdcService	Must be set to WORKFLOW_REJECT_FORM.

Example

```
IdcService=WORKFLOW_REJECT_FORM
```

```
dWfName=C10
```

```
dID=44
```

WORKFLOW_START

Description

This workflow related service starts a named workflow.

- ❖ Given a workflow name and a security group, the service starts the basic workflow.
- ❖ If this service is unable to execute, an error message is displayed to the user.

Required Parameters

These parameters must be specified:

Parameter	Description
dSecurityGroup	The security group such as PUBLIC or SECURE.
dWfName	The workflow name.
IdcService	Must be set to WORKFLOW_REJECT_FORM.

Optional Parameters

This parameter may be specified:

Parameter	Description
wfMessage	The workflow message.

Example

```
IdcService=WORKFLOW_START
dWfName=C10
dSecurityGroup=Public
```

wfMessage=Please start this workflow

Index



A

- Access Number, 5-28, 5-165
- ActiveX command utility, *See methods*, 3-4
- ADD_ALIAS, 5-8
- ADD_ARCHIVE, 5-10
- ADD_COLLECTION, 5-12
- ADD_DOC_ACCOUNT, 5-14
- ADD_DOCEXTENSION, 5-15
- ADD_DOCFORMAT, 5-16
- ADD_DOCTYPE, 5-18
- ADD_EDIT_PROVIDER, 5-19
- ADD_GROUP, 5-21
- ADD_METADEF, 5-23
- ADD_PROBLEMREPORT, 5-24
- ADD_ROLE, 5-26
- ADD_USER, 5-27
- ADD_WF_TEMPLATE, 5-32
- ADD_WORKFLOW, 5-33
- ADD_WORKFLOWALIASES, 5-35
- ADD_WORKFLOWDOCUMENT, 5-37
- ADD_WORKFLOWDOCUMENTS, 5-39
- ADD_WORKFLOWSTEP, 5-41
- Admin Server Related Services, 5-2, 5-2, 5-2
 - GET_SERVER_OUTPUT, 5-2
 - UPLOAD_NEW_COMPONENT, 5-2
- APPLET_DOCINFO, 5-45
- Attribute Name, 5-29, 5-164
- Attribute Type, 5-29, 5-165

C

- c connection_mode, 2-4
 - auto, 2-5
 - server, 2-5
 - standalone, 2-5
- calling services remotely, 2-13
- CANCEL_ARCHIVE, 5-46
- CANCEL_SEARCH_INDEX, 5-47
- CHECK_USER_CREDENTIALS, 5-48
- CHECKIN_ARCHIVE, 5-50
- CHECKIN_BYNAME, 5-57, 5-58
- CHECKIN_LIST, 5-65
- CHECKIN_NEW, 5-66
- CHECKIN_NEW_FORM, 5-73
- CHECKIN_SEL, 5-74
- CHECKIN_SEL_FORM, 5-81
- CHECKIN_UNIVERSAL, 5-82
- CHECKOUT, 5-89
- CHECKOUT_BY_NAME, 5-91
- CHECKOUT_OK, 5-93
- closeServerConnection(), 3-7
- command file syntax, 2-7
 - precedence, 2-7
 - special characters
 - , 2-8
 - #, 2-8
 - =, 2-8
 - EOD, 2-7
 - special tags, 2-8
 - IdcService=, 2-8
- command line options, 2-3

Index

- c connection_mode, 2-4
- f command_file, 2-3
- l log_file, 2-4
- u user_name, 2-4
- computeNativeFilePath(Data As String) as string, 3-9
- computeURL(Data As String, IsAbsolute As Boolean) as string, 3-10
- computeWebFilePath(Data As String) as string, 3-8
- CONFIG_INFO, 5-2, 5-94
- configuration file options, 2-5
 - ConnectionMode, 2-6
 - IdcCommandFile, 2-6
 - IdcCommandLog, 2-6
 - IdcCommandUserName, 2-6
- ConnectionMode, 2-6
 - auto, 2-6
 - server, 2-7
 - standalone, 2-7
- ConnectToServer() as boolean
 - returns, 3-6
- connectToServer() as boolean, 3-6
- Content Publisher Related Services, 5-3
 - DOC_INFO_LATESTRELEASE, 5-3
 - GET_DOC_CONFIG_INFO, 5-3
 - GET_DOCUMENT_PROBLEMREPORTS, 5-3
- CONTROL_SEARCH_INDEX, 5-95
- COPY_ARCHIVE, 5-104
- CREATE_SUBSCRIPTION_TYPE, 5-105
- creating and executing IdcCommand parameters, 2-9, 2-9
 - run IdcCommand on NT, 2-11
 - run IdcCommand on Solaris, 2-11
- CRITERIAWORKFLOW_DISABLE, 5-107
- CRITERIAWORKFLOW_ENABLE, 5-108
- Customization Related Services, 5-3
 - ADD_USER, 5-3
 - CHECKIN_UNIVERSAL, 5-3
 - CHECKOUT, 5-3
 - CHECKOUT_BY_NAME, 5-3
 - DELETE_USER, 5-3
 - EDIT_USER, 5-3
 - GET_DYNAMIC_CONVERSION, 5-4

- GET_FILE, 5-4
- GET_TABLE, 5-4
- MERGE_TABLE, 5-4
- PAGE_HANDLER, 5-4
- START_SEARCH_INDEX, 5-4
- UNDO_CHECKOUT, 5-4
- UNDO_CHECKOUT_BY_NAME, 5-4
- UPDATE_META_TABLE, 5-4
- UPDATE_OPTION_LIST, 5-4

D

- DEL_METADEF, 5-109
- DELETE, 5-124
- DELETE_ALIAS, 5-110
- DELETE_ARCHIVE, 5-111
- DELETE_BATCH_FILE, 5-113
- DELETE_BATCH_FILE_DOCUMENTS, 5-115
- DELETE_DOC, 5-117
- DELETE_DOC_ACCOUNT, 5-118
- DELETE_DOCEXTENSION, 5-119
- DELETE_DOCFORMAT, 5-120
- DELETE_DOCTYPE, 5-121
- DELETE_GROUP, 5-122
- DELETE_PROBLEMREPORT, 5-123
- DELETE_PROVIDER, 5-124
- DELETE_RESULT_TEMPLATE, 5-125
- DELETE_REV, 5-126
- DELETE_REV_EX, 5-127
- DELETE_ROLE, 5-128
- DELETE_SUBSCRIPTION_TYPE, 5-129
- DELETE_USER, 5-130
- DELETE_WF_TEMPLATE, 5-131
- DELETE_WFCONTRIBUTORS, 5-132
- DELETE_WORKFLOW, 5-134
- DELETE_WORKFLOWCRITERIA, 5-135
- DELETE_WORKFLOWDOCUMENTS, 5-136
- DELETE_WORKFLOWSTEP, 5-137
- DOC_FORMATS_WIZARD, 5-139
- DOC_INFO, 5-140
- DOC_INFO_BY_NAME, 5-141
- DOC_INFO_LATESTRELEASE, 5-142
- DOC_SUBS_LIST, 5-143

DOWNLOAD_COMPONENT, 5-2, 5-144

E

EDIT_ALIAS, 5-145
 EDIT_ARCHIVE, 5-147
 EDIT_ARCHIVEDATA, 5-148
 EDIT_DOC_FORMATS, 5-149
 EDIT_DOCEXTENSION, 5-151
 EDIT_DOCFORMAT, 5-152
 EDIT_DOCTYPE, 5-154
 EDIT_EXPORTERS, 5-155
 EDIT_GROUP, 5-157
 EDIT_METADEF, 5-158
 EDIT_ROLE, 5-161
 EDIT_USER, 5-162
 EDIT_USER_PROFILE, 5-167
 EDIT_WF_TEMPLATE, 5-169
 EDIT_WORKFLOW, 5-172
 EDIT_WORKFLOWCRITERIA, 5-207
 EDIT_WORKFLOWSTEP, 5-176
 E-mail
 of technical support, 1-5
 ENABLE_DISABLE_PROVIDER, 5-180
 EOD, 2-7
 executeCommand (Data As String), 3-5
 parameters
 Data, 3-5
 returns, 3-5
 EXPORT_ARCHIVE, 5-182

F

-f command_file, 2-3
 FORM_PROCESS, 5-185
 FORM_SUBMIT, 5-186

G

GET, 5-2
 GET_ACTIVE_WORKFLOWS, 5-187
 GET_ADD_EDIT_PROVIDER_FORM, 5-188

GET_ADMIN_PAGE, 5-191
 GET_ALIASES, 5-192
 GET_ALL_PROVIDERS, 5-193
 GET_ARCHIVECOLLECTIONS, 5-194
 GET_ARCHIVED_FILE, 5-195
 GET_ARCHIVES, 5-198
 GET_BATCH_FILE_DOCUMENTS, 5-199
 GET_BATCH_SCHEMA, 5-201
 GET_BATCH_VALUES, 5-203
 GET_BATCHFILES, 5-205
 GET_DATARESULTSET, 5-209
 GET_DOC_CONFIG_INFO, 5-210
 GET_DOC_PAGE, 5-212
 GET_DOC_SUBSCRIBERS, 5-214
 GET_DOCEXTENSIONS, 5-215
 GET_DOCFORMATS, 5-216
 GET_DOCTYPES, 5-217
 GET_DOCUMENT_PROBLEMREPORTS, 5-218
 GET_DYNAMIC_CONVERSION, 5-219
 GET_DYNAMIC_PAGE, 5-223
 GET_DYNAMIC_URL, 5-224
 GET_EXTERNAL_DOC_INFO, 5-225
 GET_EXTERNAL_HIGHLIGHT_INFO, 5-227
 GET_EXTERNAL_XML_HIGHLIGHT_INFO, 5-229
 GET_FILE, 5-231
 GET_FILE_BY_NAME, 5-235
 GET_FILELIST, 5-239
 GET_HIGHLIGHT_INFO, 5-240
 GET_METADEFs, 5-243
 GET_OPTION_LIST, 5-244
 GET_PORTAL_PAGE, 5-245
 GET_PROBLEMREPORTS_SEARCH_FORM,
 5-247
 GET_PROBLEMREPORTS_SEARCH_
 RESULTS, 5-248
 GET_PROJECT_INFO, 5-250
 GET_PROJECT_REGISTRATION, 5-251
 GET_PROVIDER_INFO, 5-252
 GET_PROXIED_ARCHIVECOLLECTIONS, 5-254
 GET_REGISTEREDPROJECTS, 5-255
 GET_REPLICATION_DATA, 5-256
 GET_RESULT_OPTIONS, 5-257
 GET_SEARCH_RESULTS, 5-258
 GET_SECURE_PAGE, 5-260

Index

GET_SELF_REGISTER_PAGE, 5-261
GET_SERVER_OUTPUT, 5-262
GET_TABLE, 5-264
GET_UPDATE_FORM, 5-265
GET_UPDATE_PROBLEMREPORT_FORM,
5-266
GET_USER_INFO, 5-267
GET_USERS, 5-268
GET_WF_TEMPLATE, 5-269
GET_WF_TEMPLATES, 5-270
GET_WORKFLOW, 5-271
GET_WORKFLOW_INFO, 5-272
GET_WORKFLOWDOCREVISIONS, 5-274
GET_WORKFLOWDOCUMENTS, 5-275
GET_WORKFLOWS, 5-276
GET_XML_HIGHLIGHT_INFO, 5-277

I

IdcCommand

- repository server Command Utility, 2-1, 2-9
 - calling services remotely, 2-13
 - command file syntax, 2-7
 - command line options, 2-3
 - configuration file options, 2-5
 - using the Stellent launcher, 2-18

IdcCommand Services, 5-1

IdcCommandFile, 2-6

IdcCommandLog, 2-6

IdcCommandUserName, 2-6

IdcCommandX-repository server ActiveX
Command Utility, 3-1

- methods, 3-4

IdcService= (*command file syntax tag*), 2-8

IMPORT_ARCHIVE, 5-279

IMPORT_DOCUMENT, 5-282

init as boolean (*IntradocDir As String*)
parameters

- UserName, 3-4

init as boolean (*StellentDir As String*)

parameters

- StellentDir, 3-4

Internet website of technical support, 1-6

IntradocDir As String (*init as boolean*), 3-4

L

-l log_file, 2-4

LOAD_GLOBALINCLUDES, 5-284

LOAD_RESOURCE_FILE, 5-285

LOGIN, 5-288

M

MERGE_TABLE, 5-290

methods (*ActiveX command utility*), 3-4

- closeServerConnection(), 3-7

- computeNativeFilePath(Data As String) as
string, 3-9

- computeURL(Data As String, IsAbsolute As
Boolean) as string, 3-10

- computeWebFilePath(Data As String) as
string, 3-8

- connectToServer() as boolean, 3-6

- executeCommand (Data As String), 3-5

- init as boolean (IntradocDir As String), 3-4

N

NOTIFY_CHANGE, 5-294

NOTIFY_CONTRIBUTOR, 5-295

O

Overview

- Audience, 1-2

- Conventions, 1-2

- Stellent Product Distinctions, 1-3

P

PAGE_HANDLER, 5-296

Personalization Related Services, 5-5

- GET_PORTAL_PAGE, 5-5

GET_SELF_REGISTER_PAGE, 5-5
 PNE_GET_SEARCH_RESULTS, 5-5
 PNE_SAVE_QUERY, 5-5
 PNE_UPDATE_PERSONAL_URLS, 5-5
 PNE_UPDATE_PORTAL_INFO, 5-5
 PING_SERVER, 5-298
 PNE_GET_SEARCH_RESULTS, 5-300
 PNE_SAVE_QUERY, 5-301
 PNE_UPDATE_PERSONAL_URLS, 5-302
 PNE_UPDATE_PORTAL_INFO, 5-304
 precedence, 2-7
 Problem Report Related Services, 5-6
 ADD_PROBLEMREPORT, 5-6
 DELETE_PROBLEMREPORT, 5-6
 GET_DOCUMENT_PROBLEMREPORTS,
 5-6
 GET_UPDATE_PROBLEMREPORT_FORM,
 5-6
 PROBLEMREPORT_INFO, 5-6
 RESEND_PROBLEMREPORT, 5-6
 UPDATE_PROBLEMREPORT, 5-6
 PROBLEMREPORT_INFO, 5-305
 Provider Related Services, 5-7
 ADD_EDIT_PROVIDER, 5-7
 DELETE_PROVIDER, 5-7
 ENABLE_DISABLE_PROVIDER, 5-7
 GET_ADD_EDIT_PROVIDER_FORM, 5-7
 GET_ALL_PROVIDERS, 5-7
 GET_PROVIDER_INFO, 5-7
 NOTIFY_CHANGE, 5-7
 REQUEST_SECURITYINFO, 5-7
 TEST_PROVIDER, 5-7

Q

QUERY_DOC_ACCOUNTS, 5-306
 QUERY_GROUP, 5-307
 QUERY_USER_ATTRIBUTES, 5-308

R

REGISTER_IMPORTER, 5-309
 REGISTER_USER, 5-311

REMOVE_COLLECTION, 5-313
 REMOVE_EXPORTER, 5-315
 REMOVE_IMPORTER, 5-316
 REQUEST_SECURITYINFO, 5-317
 RESEND_PROBLEMREPORT, 5-318
 RESUBMIT_FOR_CONVERSION, 5-319
 REV_HISTORY, 5-320
 run IdcCommand on NT, 2-11
 run IdcCommand on Solaris, 2-11

S

SAVE_GLOBALINCLUDES, 5-321
 SELECTDOC, 5-322
 START_SEARCH_INDEX, 5-323
 SUBMIT_HTML_FORM, 5-325
 SUBSCRIBE, 5-327
 SUBSCRIBE_DOC_USER, 5-329
 SUBSCRIBE_EX, 5-331
 SUBSCRIPTION_LIST, 5-333
 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
 TEST_PROVIDER, 5-334

U

-u user_name, 2-4
 UNDO_CHECKOUT, 5-335
 UNDO_CHECKOUT_BY_NAME, 5-337
 UNSUBSCRIBE, 5-339

Index

UNSUBSCRIBE_FROM_LIST, 5-340
UNSUBSCRIBE_FROM_LIST_EX, 5-341
UPDATE_DOCINFO, 5-343
UPDATE_DOCINFO_BYFORM, 5-346
UPDATE_DOCINFO_WITH_FILE, 5-349
UPDATE_META_TABLE, 5-352
UPDATE_OPTION_LIST, 5-354
UPDATE_PROBLEMREPORT, 5-356
UPDATE_RESULT_TEMPLATE, 5-358
UPDATE_SUBSCRIPTION_NOTIFY, 5-360
UPDATE_SUBSCRIPTION_TYPE, 5-361
UPDATE_SUBSCRIPTION_USED, 5-363
UPDATE_USEROPTION_LIST, 5-365
UPLOAD_NEW_COMPONENT, 5-367
Using Services, 4-1
using the Stellent Launcher, 2-18

V

VIEW_DOC, 5-368

W

Website for technical support, 1-6
WORK_IN_PROGRESS, 5-370
WORKFLOW_APPROVE, 5-371
WORKFLOW_CANCEL, 5-372
WORKFLOW_CHECKIN, 5-373
WORKFLOW_REJECT, 5-376
WORKFLOW_REJECT_FORM, 5-377
WORKFLOW_START, 5-378