

ImageNow Message Agent

Getting Started Guide

ImageNow Version: 6.7.x

© 2014 Perceptive Software. All rights reserved

CaptureNow, ImageNow, Interact, and WebNow are trademarks of Lexmark International Technology SA, registered in the U.S. and other countries. Perceptive Software is a stand-alone business unit within Lexmark International Technology SA. All other brands and product names mentioned in this document are trademarks or registered trademarks of their respective owners. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or any other media embodiments now known or hereafter to become known, without the prior written permission of Perceptive Software.

Table of Contents

Getting Started with Message Agent	5
What is Message Agent?.....	5
Message Agent Overview.....	6
Development Requirements.....	6
Documentation	7
Message Agent web services model	7
<i>The Access Service endpoint</i>	<i>7</i>
<i>The Document Service endpoint</i>	<i>7</i>
<i>The Folder Service endpoint.....</i>	<i>7</i>
<i>The Form Service endpoint</i>	<i>7</i>
<i>The Task Service endpoint</i>	<i>7</i>
<i>The Workflow Service endpoint.....</i>	<i>8</i>
Prepare your application for Message Agent	9
Generate the client proxy code	9
<i>Reference the Web Services</i>	<i>10</i>
Access Web Service	10
Log in to Message Agent	10
Verify a valid ImageNow user	11
Audit a user	11
Log out of Message Agent.....	11
INOW_BOOLEAN_RESPONSE supporting data type	11
INOW_CONTEXT supporting data type	12
Document Web Service.....	13
Search for documents.....	13
Get information about a document	13
View a document.....	14
Folder Web Service.....	14
Create a folder.....	14
Retrieve a folder	14
Update a folder.....	14
Form Web Service.....	15
Add a form.....	15
View a form.....	15

Update a form	15
Publish a form	15
Task Web Service	15
Create a task	15
Complete a task	16
Return assigned tasks	16
Retrieve task properties	16
Retrieve task reasons	16
Workflow Web Service	16
Add a document to workflow	16
Search for workflow queue item information	16
Remove a document from workflow	17
Error Messages	17
Access Web Service	18
Document Web Service	18
Folder Web Service	21
Form Web Service	22
Task Web Service	23
Workflow Web Service	24
General Error Messages	25
Index	26

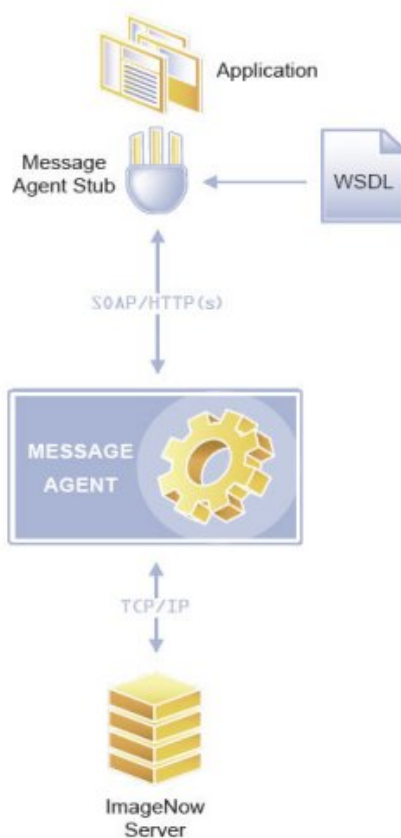
Getting Started with Message Agent

This document introduces developers to Message Agent. It provides an overview of Message Agent requirements and its web services. You will learn about the high level architecture of the web services in Message Agent for writing your business applications, and how to test Message Agent using a third party utility. You will also read about troubleshooting error handling operations, and how to interpret error codes encountered when running and using Message Agent.

What is Message Agent?

Message Agent is a middle-tier web service that provides communication over a network between ImageNow software and third-party applications. Message Agent exports a rich set of functions as web services enabling you, as the developer, to embed ImageNow functionality, such as document management and workflow, directly into your business application.

The following figure shows how a client business application makes a call to Message Agent using standards-based Simple Object Access Protocol (SOAP) Extensible Markup Language (XML), over HTTP/HTTPs transport protocol. The Message Agent then uses TCP/IP to communicate with the ImageNow Server returning web service request and response messages for you to incorporate into your business application. By using TCP/IP, Message Agent can install anywhere on your network. The WSDL files and SOAP can work with third party applications in any programming language.



Message Agent Overview

To embed this Message Agent functionality, you create applications using generated client proxy code stubs in a development tool from the WSDL (Web Services Description Language) files and the XML schema files provided with Message Agent. WSDL is the standard format for describing a web service. Expressed in XML, a WSDL definition describes how to access a web service and what operations the web service can perform. Message Agent supports WSDL, version 1.1 or higher.

You can write business logic that uses methods and functions in these client proxy code stubs. The resulting Message Agent client business applications communicate, using the SOAP libraries in their toolkits, with Message Agent over HTTP or HTTPS. Message Agent communicates with ImageNow Server over TCP/IP.

SOAP is a lightweight XML-based messaging protocol used to encode the information in web service request and response messages before sending them over a network. SOAP messages are independent of any operating system or protocol and can be transported using a variety of Internet protocols, including SMTP, MIME, and HTTP. Message Agent supports SOAP version 1.1 and higher.

Message Agent provides the required files for the ImageNow Server at runtime. You access ImageNow functionality through the web services. Each web service uses WSDL operations to expose select ImageNow functions that let you perform such actions as storing, copying, and getting status on documents in the ImageNow repository. Message Agent then makes those functions available to third-party applications.

You use Message Agent and a web services toolkit to generate client-side libraries from the XSD files and WSDL files (tested against the WS-I Basic Profile). WS-I stands for Web Services Interoperability. You can embed ImageNow functionality into applications with a minimal knowledge of SOAP and web services technologies.

To ensure maximum interoperability, Message Agent web services follow the document/literal wrapped style. A WSDL binding describes how the service is bound to a messaging protocol. A SOAP binding can also have an encoded use or a literal use, and the document/literal wrapped style is one of four available encoded or literal styles available.

Message Agent generates all operations as classes, functions, methods, subroutines or other types of routines as your development tool requires. The code generator also provides the necessary variable declarations for your development tool. You can then write business logic that includes the variables to incorporate the ImageNow functionality into your application. To ensure the best interoperability, maximum performance, and minimal errors, the web services that Message Agent exposes conform to the standards defined by the World Wide Web Consortium (WC3) and WS-Interoperability (WS-I).

Development Requirements

When developing applications for Message Agent, you must use a language that provides a WSDL 1.1-compliant code generator. Most third-party development tools, such as the J2EE and .NET frameworks, provide compilers for WSDL to Java, C#, and C++.

Any third-party software that is capable of both sending and receiving SOAP/XML messages, regardless of operating system and development platform, should be capable of interfacing with Message Agent.

Documentation

The documentation provided with Message Agent includes online help, XML documentation, and Java code samples.

Help files are located in the `\inserver6\help` directory. To view help, navigate to `\inserver6\help`, and then double-click **Start_MessageAgent_Help.htm**. You can also use the Perceptive Software website at www.perceptivesoftware.com. This website always contains the most current version of the online help.

The `\inserver6\MessageAgent\xmldocs` directory provides XML documentation, generated from XML and XSD files in Message Agent. Each web service has its own subdirectory. For example, to review the schema for `imagenow_document.xsd`, double-click **imagenow_document.html** in `\inserver6\MessageAgent\xmldocs\DOCUMENT_SERVICE`.

Message Agent web services model

The benefit of using standard web protocols is application interoperability. Message Agent uses three protocols – XML, SOAP and WSDL– to provide access to its web services.

The Access Service endpoint

The Access Service endpoint exports basic ImageNow authentication and authorization operations. You use access operations for logging in to ImageNow Server and opening a session, for logging out and closing a session, and for validating a user. You always begin with the start session operation and finish with the end session operation.

The Document Service endpoint

The Document Service endpoint exports basic ImageNow document management operations. You use file operations including copying documents, moving documents, storing documents, and deleting documents, as well as control operations for document locking and unlocking actions. There are also Metadata operations, including actions to manage keywords and document properties.

The Folder Service endpoint

The Folder Service endpoint exports folder operations you use to create a folder, identify a document by its document ID or a unique set of document key values, and identify the folder by its unique ID or a combination of its name and type. Search operations using a hierarchical to a drawer, document, folder, or shortcut are also available. Operations to return a collection of objects containing attributes/metadata associated with all active folders within ImageNow are also available.

The Form Service endpoint

The Form Service endpoint exports form management operations. You use operations to add, update and publish forms to ImageNow Server or ImageNow Form Server. There are also operations representing the state of a form as active or inactive, and properties for forms such as ID's of the custom properties associated with the form.

The Task Service endpoint

The Task Service endpoint exports task management operations. You use operations to create and complete tasks, retrieve all tasks assigned to the current user, retrieve properties for a specific task, and

add comments to a task. Operations to retrieve associated action and return reasons for tasks within ImageNow are also available.

The Workflow Service endpoint

The Workflow Service endpoint exports workflow operations for creating queue items, getting queue items, and deleting queue items. There are a set of operations for routing workflow items and obtaining document status as well as properties operations, which includes actions for getting a queue list and a user list.

About the WSDL and XSD files

Message Agent defines web services as WSDL operations. Generating client proxy code stubs from the WSDL file creates a class for each web service operation. As the developer, you write the code to use the corresponding class.

You use your development tool to generate web services into client proxy code stubs. Each tool varies according to the way it organizes each service. Each web service contains an input (request) and output (response) operation. These operation elements reside in the corresponding WSDL and XSD files for each web service you install with Message Agent. You create a corresponding routine in your client proxy code stub for each operation. The code in the corresponding routine must contain the logic to send a request to Message Agent and the logic to handle the response.

Many of the operations have supporting data types that Message Agent defines in a web services XML Schema file. Data types common to more than one operation appear in the `imagenow_commonTypes.xsd` file.

Web Service Definition Language (WSDL) files

After you install Message Agent, the WSDL files are located in `\inserver6\MessageAgent\wsdl`.

WSDL	Description
<code>imagenow_access.wsdl</code>	This file defines the basic authentication and authorization operations for accessing ImageNow using Message Agent.
<code>imagenow_document.wsdl</code>	This file defines the basic document management operations for use with Message Agent.
<code>imagenow_folder.wsdl</code>	This file defines the folder operations for use with Message Agent.
<code>imagenow_form.wsdl</code>	This file defines the form management operations for use with Message Agent.
<code>imagenow_license.wsdl</code>	This file defines the basic licensing operations for use with Message Agent.
<code>imagenow_services.wsdl</code>	This file consolidates the individual WSDL files into a single file to simplify the process of creating client stubs for all the services at the same time.
<code>imagenow_task.wsdl</code>	This file defines the task operations for use with Message Agent.
<code>imagenow_workflow.wsdl</code>	This file defines the workflow operations for use with Message Agent.

XML schema definition (XSD) files

After you install Message Agent, the XSD schema files are located in `\inserver6\MessageAgent\wsdl`.

XSD	Description
<code>imagenow_access.xsd</code>	This file describes the access data types for <code>imagenow_access.wsdl</code> using XML schema structures.
<code>imagenow_commonTypes.xsd</code>	This file describes the data types common to more than one of the WSDL files.
<code>imagenow_document.xsd</code>	This file describes the document data types for <code>imagenow_document.wsdl</code> using XML schema structures.
<code>imagenow_folder.xsd</code>	This file describes the folder data types for <code>imagenow_folder.wsdl</code> using XML schema structures.
<code>imagenow_form.xsd</code>	This file describes the form data types for <code>imagenow_form.wsdl</code> using XML schema structures.
<code>imagenow_license.xsd</code>	This file describes the licenses needed to interact with ImageNow for the <code>imagenow_license.wsdl</code> using XML schema structures.
<code>imagenow_task.xsd</code>	This file describes the task data types for <code>imagenow_task.wsdl</code> using XML schema structures.
<code>image_workflow.xsd</code>	This file describes the workflow data types for <code>imagenow_workflow.wsdl</code> using XML schema structures.

Prepare your application for Message Agent

This section explains how to generate the client proxy code in your development tool for the web services.

Generate the client proxy code

To generate client proxy (stub) code in your development tool or for your preferred programming language, you must install a corresponding SOAP toolkit. The SOAP web services toolkit uses the XML definitions in the WSDL and XSD files to generate code in your development tool and in the language you choose. You can then use the interfaces (classes, functions, methods, and subroutines) that the web services toolkit generates to write the business logic for your client application.

To generate stubs for Access, Document, Folder, Form, Task, and Workflow services simultaneously, use `imagenow_services.wsdl`. This file is a composite WSDL comprised of individual WSDL files. If you do not use this file and you use the individual WSDL files directly to generate the stubs, you may need to remove the comments around the service element. Comments surround this element to prevent duplicate service declarations with the same service defined in the composite WSDL file.

The capabilities of your web services toolkit determine the allowable content of the client proxy code stubs. The libraries associated with each web services toolkit also handle the process of serializing language specific (Java, C++, C#, Visual Basic, and more) data types from your Message Agent client requests to SOAP-XML and then de-serializing the Message Agent SOAP-XML responses.

Reference the Web Services

In your code, use the client stub generated from the service to log in and out of ImageNow through Message Agent. When you run a web service toolkit, it asks for a URL or path to the web service you want to add. The port in the URLs provided for each service is the default port that the Message Agent installer assigned. If you change the port, you will have to use that new port in the URLs below. For more information on changing the port settings, refer to “Server port settings” section of this guide.

In your code, use the client proxy code stubs generated from the web service you will add, as shown in the following examples:

Access Service

```
http://<hostname>:6070/imagenow/services/1.0/access_service/WSDL
```

Document Service

```
http://<hostname>:6070/imagenow/services/1.0/document_service/WSDL
```

Folder Service

```
http://<hostname>:6070/imagenow/services/1.0/folder_service/WSDL
```

Form Service

```
http://<hostname>:6070/imagenow/services/1.0/form_service/WSDL
```

Task Service

```
http://<hostname>:6070/imagenow/services/1.0/task_service/WSDL
```

Workflow Service

```
http://<hostname>:6070/imagenow/services/1.0/workflow_service/WSDL
```

To read more about each web service, refer to the individual web service sections in this guide.

Access Web Service

The Access Service exports basic ImageNow authentication and authorization operations. You use access operations for logging in to ImageNow Server and opening a session, for logging out and closing a session, and for validating a user. You always begin with the start session operation and finish with the end session operation.

For a complete list of Access Web Service operations, refer to the `imagenow_access.wsdl` file. To locate the `imagenow_access.wsdl` file, navigate to `\inserver6\MessageAgent\wsdl` directory on the server computer.

Log in to Message Agent

Use the input parameters in the `ACCESS_SESSION_BEGIN_USING_PASSWORD` operation to log in to ImageNow through Message Agent with a valid ImageNow user name and password. You must pass the correct operation parameters to log in to the ImageNow Server and return the `LOGIN_OBJECT`, which establishes a session for the authenticated user.

When the Message Agent returns `LOGIN_OBJECT`, this object contains authentication parameters for the Message Agent session. You use the `INOW_CONTEXT` parameter in the `LOGIN_OBJECT` in almost every operation throughout the entire session.

Two user login methods are available for starting a Message Agent session: Individual or Bridge. To start a Message Agent session, you must be a valid ImageNow user with privileges. Users of Message Agent either log in individually or as a bridge that represents several users. Users who log in through a bridge can be audited to track their activities.

To log in as an individual user, assign any number for the APP_CONTEXT parameter when starting the session. To log in as a bridge user, assign the number 1 to the APP_CONTEXT parameter when starting the session. To audit users making operation input requests through a bridge user login, add a second 1 to the APP_CONTEXT parameter for each individual web service operation call you want to audit, which makes the value of the parameter appear as 11. This field can be equal to 1 or 11 only.

Verify a valid ImageNow user

Use the input parameters defined in the ACCESS_USER_CHECK operation to determine whether a provided user name is a valid ImageNow user. If the request fails, Message Agent returns an exception.

Audit a user

You can audit a user who is coming through a bridge session by setting the APP_CONTEXT parameter of the INOW_CONTEXT element to a value of 11. If you want to audit a specific user action and apply security for the user being audited separately from applying security of the bridge, set the AUDIT_USER parameter to the user name of the user you want to audit. For more details on the APP_CONTEXT parameter settings, refer to the “INOW_CONTEXT supporting data type” section of this guide.

Log out of Message Agent

Use the input parameters defined in the ACCESS_SESSION_END operation to end an existing user session. If the request fails, Message Agent returns an exception.

Warning When the bridge ends its session, if the user is logged on as a bridge, Message Agent disconnects all users connecting through that bridge.

INOW_BOOLEAN_RESPONSE supporting data type

You can receive several types of output from Access Service operations. The majority of operations return a Boolean value defined by the INOW_BOOLEAN_RESPONSE supporting data type. This response communicates to the client application whether the request succeeds or fails. The RESPONSE_CODE and RESPONSE_MESSAGE elements of INOW_BOOLEAN_RESPONSE output parameter will only be populated if the INOW_BOOLEAN value indicates false (0).

INOW_CONTEXT supporting data type

After you log in to Message Agent, the INOW_CONTEXT element values move back and forth with every operation call. You use the INOW_CONTEXT session for every operation call until the user logs out of Message Agent. This element provides the login information so that you do not need to provide it with each operation call. The SESSION_STRING parameter, which represents the user's context, is important. The following table shows the INOW_CONTEXT element and its supporting data elements, one of which includes the SESSION_STRING element.

INOW_CONTEXT				
Parameter Name	Description	XML Schema Data	Example Value	Value
LOCALE	A placeholder for internationalization or browser location value.	Token	US_en	
APP_CONTEXT	<p>Identifies the type of user.</p> <p>Bridge user. The number 1 indicates that the user is a bridge user. A bridge user provides session authentication and makes web service calls on behalf of audit users. A bridge user must first authenticate with the ImageNow system using ACCESS_SESSION_BEGIN_USING_PASSWORD before audit users can begin making web service calls through the bridge user.</p> <p>Audit user. The number 11 indicates that the user is an audit user. Audit users obtain session authentication through their bridge user. If the user is an audit user, Message Agent evaluates the AUDIT_USER field to verify that it contains a valid ImageNow user name.</p> <p>Individual user. Any number other than 1 or 11 indicates that the user is an individual user. Individual users authenticate directly with the ImageNow system using ACCESS_SESSION_BEGIN_USING_PASSWORD.</p>	Token	11	
SESSION_STRING	The string ImageNow returns when the user first logs in. Every following call has to carry a valid session string.	Token	1108656262_000395000208	Maximum field length = 23

INOW_CONTEXT				
MESSAGE_AGENT_SERVER_NAME	The host name of the Message Agent server.	Token	MAServer	
INOW_USER_NAME	The name of the user in ImageNow.	USER_NAME_TYPE	ImageNowUser1	Maximum field length = 256
AUDIT_USER	The user name of the user coming through the bridge user account. Security permissions and auditing are conducted on this user name. For the audit user to authenticate and be allowed to make web service calls, this must be a valid ImageNow user name.	USER_NAME_TYPE	ImageNowUser1	Maximum field length = 256

Document Web Service

The Document Service provides operational calls to use as you develop your business application for ImageNow document management functionality. Use these operations to locate a document stored in ImageNow, get information about a document, and return a document based on its metadata.

For a complete list of Document Web Service operations, refer to the `imagenow_document.wsdl` file. To locate the `imagenow_document.wsdl` file, navigate to `\inserver6\MessageAgent\wsdl` directory on the server computer.

Search for documents

Use the input parameters in the `DOCUMENT_SEARCH_QUERY` operation to search ImageNow for documents. This operation searches the ImageNow repository for documents meeting the specified search query parameter criteria, such as document key values. You need the `WHERE` clause of the query to conduct the search. Parameters specify how to return the document information and which fields to return to the calling client application.

The output returns `DOCUMENT_OBJECT` with information requested by the input parameters. The `DOCUMENT_OBJECT` is a logical representation of the metadata that describes an ImageNow document. If the request fails, Message Agent returns an exception.

You must have a valid ImageNow Content Server license to conduct full-text keyword queries using this operation. ImageNow Content Server collects text data and creates searchable indexes based on the content of each document page. Using a Content Server search, you can search for specific keywords or phrases. Your search can contain wildcards to represent one or more characters. The server displays the document pages that contain matches for the search and highlights the matching keywords within the documents. Content Server lets you perform various types of full text searches, including fuzzy, phonetic, stemming and synonym, proximity, and relevance ranking searches.

Get information about a document

Use the input parameters in the `DOCUMENT_INFO_GET` operation to get information, such as the total number of pages, for a specific document. The input parameters specify what to return to the client application. The output parameters return the `DOCUMENT_OBJECT` with information requested by the

input parameters. The DOCUMENT_OBJECT is a logical representation of the metadata describing an ImageNow document. If the request fails, Message Agent returns an exception.

View a document

Use the input parameters in the DOCUMENT_RETURN operation to return a document object. If you don't specify page numbers, this operation returns only the first distinct page of the document. If you provide a value of -1 for the page number, Message Agent returns all pages of the document. When the request is successful, the output parameters return the document object and information about the document object. If the request fails, Message Agent returns an exception. If you specify PDF for the DOC_OUTPUT_FORMAT when the file type of cannot be converted to PDF, Message Agent returns an exception.

Important Use this operation for SOAP toolkits that do not support the SOAP with Attachments standards, or if you need to return multiple, separate, page files from a single request. The .NET framework does not support SOAP with Attachments. Otherwise, use the DOCUMENT_RETURN_SWA operation.

Folder Web Service

The Folder Service provides operations calls to use as you develop your business application with folders. You can use operations to create folders, get attributes/metadata associated with a folder, and identify a folder by its document ID or its unique set of document key values.

For a complete list of Folder Web service operations, refer to the imagenow_folder file. To locate the **imagenow_folder.wsdl** file, navigate to `\inserver6\MessageAgent\wsdl` directory on the server computer.

Create a folder

Use the input parameters in the FOLDER_CREATE operation to add a new folder available for association with various ImageNow documents. The custom properties affiliated with the new folder are based upon the specified folder type, FOLDER_TYPE_NAME. This operation returns an object that indicates the success or failure of the request. If the request fails, Message Agent returns an exception.

Retrieve a folder

Use the input parameters in the FOLDER_GET operation to return an object containing all attributes/metadata associated with the provided folder. The folder is identified within the request through either its unique ID or a combination of its name and folder type. If the request fails, Message Agent returns an exception.

Use the input parameters in the FOLDER_GET_PATH_FOR_OBJECT operation to search for a document, folder, shortcut, or drawer based on its unique ID and return the full hierarchical folder path. If the request fails, Message Agent returns an exception.

Update a folder

Use the input parameters in the FOLDER_UPDATE operation to update the attributes associated with a given ImageNow folder. Once the folder has been identified through either its unique ID or its name / folder type combination, various elements associated with the folder can be updated, such as its name, active status, or its associated custom property value. The output returns an object that indicated the success or failure of the folder update. If the request fails, Message Agent returns an exception.

Form Web Service

The Form Service provides operations to enhance your business application through the use of forms. You can use operations for adding, viewing, updating, and publishing a form to ImageNow Server or ImageNow Form Server.

For a complete list of Form Web Service operations, refer to the `imagenow_form.wsdl` file. To locate the `imagenow_form.wsdl` file, navigate to `\inserver6\MessageAgent\wsdl` directory on the server computer.

Add a form

Use the input parameters in the `FORM_ADD` operation to add a form to ImageNow Server. The output returns the unique ID of the form. If the request fails, Message Agent returns an exception.

View a form

Use the input parameters in the `FORM_GET_ALL` operation to add a form to ImageNow Server. The output returns an array of information for a form including its unique ID, the name of the form, the form description, and a parameter indicating whether the form is active or inactive. If the request fails, Message Agent returns an exception.

Update a form

Use the input parameters in the `FORM_UPDATE` operation to update a form in ImageNow. The `FORM_UPDATE_PRESENTATION` operation is available to update a form presentation in ImageNow. Each of these operations returns a parameter indicating whether the form is active or inactive. If the request fails, Message Agent returns an exception.

Publish a form

Use the input parameters in the `FORM_PUBLISH_BY_ID` operation to publish a form to ImageNow Server or the ImageNow Form Server. The output returns properties for the form, including the `QUEUE_NAME` and `QUEUE_ID` of the workflow queue of the form for identification by the Forms Server. A parameter representing the URL for opening forms on ImageNow Form Server is also part of this operation. If the request fails, Message Agent returns an exception.

Task Web Service

The Task Service contains operations for task management. Use these operations to create a task, complete a task, and return assigned tasks. There are also operations available for retrieving the properties and reasons associated with tasks.

For a complete list of Task Web Service operations, refer to the `imagenow_task` file. To locate the `imagenow_task.wsdl` file, navigate to `\inserver6\MessageAgent\wsdl` directory on the server computer.

Create a task

Use the input parameters in the `TASK_CREATE` operation to create a task. This operation returns an array of task identifiers. If the request fails, Message Agent returns an exception.

Complete a task

Use the input parameters in the TASK_COMPLETE operation to complete a task. Note you cannot use this operation to complete retention approval tasks or a signature required task if the location for that task is set to the page with a visual representation. If the request fails, Message Agent returns an exception.

Return assigned tasks

Use the input parameters in the MY_ASSIGNED_TASKS operation to display a list of tasks assigned to the current user. The operation returns an array of assigned tasks. When the request is successful, the output parameters return the task object and information associated with that object. If the request fails, Message Agent returns an exception.

Retrieve task properties

Use the input parameters in the TASK_GET_PROPERTIES operation to retrieve the properties associated with a specific task. When the request is successful, the output parameters return the task properties object and information associated with that object. If the request fails, Message Agent returns an exception.

Retrieve task reasons

Use the input parameters in the TASK_GET_REASONS operation to retrieve the action and return reasons associated with a task template. When the request is successful, the output parameters return the task reasons object and information associated with that object. If the request fails, Message Agent returns an exception.

Workflow Web Service

The Workflow Service provides a variety of operational calls you can use to return information on documents in workflow. Operations to manage queue activities are available for use, as well as operations to obtain a queue list, and for a specific workflow queue.

For a complete list of Workflow Web Service operations, refer to the imagenow_workflow.wsdl file. To locate the **imagenow_workflow.wsdl** file, navigate to `\server6\MessageAgent\wsdl` directory on the server computer.

Add a document to workflow

Use the input parameters in the WORKFLOW_QITEM_SET operation to set a document as an item to be managed in workflow. The parameters enable you to specify a workflow queue name, and use the QITEM_SET_MODE element to select which parameters to identify the entity when adding to workflow, including DOC_KEYS, DOCID, PROJECTID, FOLDERID, FOLDER_TYPE_NAME, and PROJECT_TYPE_NAME. If the request fails, Message Agent returns an exception.

Search for workflow queue item information

Use the input parameters in the WORKFLOW_QITEM_GET operation to return the information on a specific workflow item. The parameters enable you to specify how to return the document information and which data fields to provide to the calling client application. If the request fails, Message Agent returns an exception.

Use the `WORKFLOW_QITEM_GET_INFO` to obtain information about a particular document in a particular workflow queue. The output parameters return the `WORKFLOW_QITEM` object, which includes both workflow-specific state information as well as an embedded `DOCUMENT_OBJECT` or `PROJECT` object (depending upon the nature of the workflow queue item). If the request fails, Message Agent returns an exception.

Use the input parameters in the `WORKFLOW_QITEM_GET_LIST` operation to get a list of workflow queue items. If the request fails, Message Agent returns an exception.

Remove a document from workflow

Use the input parameters defined in the `WORKFLOW_QITEM_DELETE` operation to delete a workflow item. This operation does not delete the document from the Object Storage Manager (OSM). If the request fails, Message Agent returns an exception.

Error Messages

As you develop your business application using Message Agent, you may encounter situations when an operation fails causing an error condition. To help you manage these situations, this section includes the operation name, the operation ID, error codes, and error messages for each web service.

The `INOW_EXCEPTION` response is the normal output for any operation that fails. This supporting data type is an object with its own parameters. These parameters specify the error ID, `ERROR_CODE`, and `ERROR_MESSAGE`. You can use this output to troubleshoot operation failure. All operations can return `INOW_EXCEPTION` instead of, or along with, the expected output when an operation fails.

An `INOW_EXCEPTION` consists of the following elements:

Element	Description
ID	A number that represents the ID of the operation with the exception.
ERROR_CODE	A unique code for the associated process error or failure.
ERROR_MESSAGE	A detailed description of the reason for the process error or failure. It is associated with the accompanying <code>ERROR_CODE</code> value.

Whenever Message Agent returns an `INOW_EXCEPTION`, it returns the three elements shown in the table above to the client application. These exception errors appear in ID order, where ID is a unique number assigned to each operation. After you identify the operation, the `ERROR_CODE` and `ERROR_MESSAGE` provide more detail about the cause of the error.

Access Web Service

The following tables provide the IDs, error codes, and error messages that can be returned as exceptions for operations in the Access Web service.

Access Service Operation and ID

Operation	ID
ACCESS_SESSION_BEGIN_USING_PASSWORD	10000
ACCESS_SESSION_END	10001
ACCESS_USER_CHECK	10002

Access Service Error Codes and Messages

Error Code	Error Message
1001	Authentication Failure.
1002	The user has not been authenticated.
1003	The User ID is invalid or not in the correct format.
1004	The user name is invalid or not in the correct format.
1005	The user type is invalid.
1006	A blank password is not allowed.
1007	An audit user already exists.
1008	A different type of user already exists.

Document Web Service

The following tables provide the IDs, error codes, and error messages that can be returned as exceptions for operations in the Document Web service.

Document Service Operation and ID

Operation	ID
DOCUMENT_MOVE	20000
DOCUMENT_COPY	20001
DOCUMENT_STORE	20002
DOCUMENT_DELETE	20003
DOCUMENT_NOTIFY_NEW	20004
DOCUMENT_KEYWORDS_GET	20005
DOCUMENT_UNLOCK_ALL	20006
DOCUMENT_LOCK	20007

Operation	ID
DOCUMENT_UNLOCK	20008
DOCUMENT_INFO_GET	20009
DOCUMENT_KEYWORDS_SET	20010
DOCUMENT_KEYWORDS_DELETE	20011
DOCUMENT_SEARCH_QUERY	20012
DOCUMENT_STORE_SWA	20013
DOCUMENT_RETURN	20014
DOCUMENT_RETURN_SWA	20015
DOCUMENT_DRAWER_GET_LIST	20016
DIGITAL_SIGNATURE_REASON_GET_ALL	20017
DOCUMENT_SIGN	20018
DOCUMENT_SIGNATURE_VERIFY	20019
DOCUMENT_SIGNATURE_VERIFY_ALL	20020
DOCUMENT_SIGNATURE_GET_LIST	20021
DOCUMENT_SIGNATURE_VOID	20022
CUSTOM_PROPERTY_GET	20023
CUSTOM_PROPERTY_GET_ALL	20024
DOCUMENT_UPDATE_CUSTOM_PROPERTIES	20025
PROJECT_TYPE_GET	20026
PROJECT_TYPE_GET_ALL	20027
PROJECT_CREATE	20028
PROJECT_UPDATE	20029
PROJECT_DELETE	20030
PROJECT_GET	20031
PROJECT_GET_ALL	20032
DOCUMENT_ADD_TO_PROJECT	20033
DOCUMENT_REMOVE_FROM_PROJECT	20034
DOCUMENT_TYPE_GET	20035
DOCUMENT_TYPE_GET_ALL	20036
DOCUMENT_TYPE_GROUP_UPDATE	20037
DOCUMENT_TYPE_GROUP_GET	20038
DOCUMENT_TYPE_GROUP_GET_ALL	20039
DOCUMENT_VERSION_CONTROL_ADD	20040
DOCUMENT_VERSION_CONTROL_REMOVE	20041

Operation	ID
DOCUMENT_CHECK_IN	20042
DOCUMENT_CHECK_IN_FILE	20043
DOCUMENT_CHECK_OUT	20045
DOCUMENT_CHECK_OUT_FILE	20046
DOCUMENT_UNDO_CHECK_OUT	20048
DOCUMENT_SET_VERSION_PUBLIC	20049
DOCUMENT_SET_VERSION_PRIVATE	20050
DOCUMENT_GET_VERSION_LIST	20051
DOCUMENT_GET_VERSION_HISTORY	20052
DOCUMENT_SET_VERSION_COMMENTS	20053
DOCUMENT_VERSION_CONTROL_PROMOTE	20054
DIGITAL_SIGNATURE_REASON_GET_LIST	20055
DOCUMENT_RELATED_WORKSHEET_RESPONSE	20056
DOCUMENT_RETURN_SEND_TO_USER	20057
PROJECT_GET_DOCUMENTS	20058
DOCUMENT_SUBMIT_TO_CONTENT	20059
IN_VIEW_GET	20060
IN_VIEW_RUN	20061
INOW_USER_GET_INFO	20062
INOW_USER_GET_GROUPS	20063
SEND_EXTERNAL_MESSAGE	20064
GET_UNIQUE_ID	20070
DOC_SET_NAME	20071

Document Service Error Codes and Messages

Error Code	Error Message
2002	Message Agent cannot get the document ID using the supplied document keys.
2003	Message Agent cannot add the document to version control.
2004	Message Agent cannot get server-side temporary directory for the user.
2005	Message Agent found nothing before timeout.
2006	You can only link by document keys.
2007	No document management license.

Error Code	Error Message
2008	Could not sign a document which is checked out.
2009	One or more target custom properties are not valid for the specified target document type.
2010	No view matches the specified search parameters.
2011	Custom property value is invalid.
2012	The password is invalid.

Folder Web Service

The following tables provide the IDs, error codes, and error messages that can be returned as exceptions for operations in the Folder Web service.

Folder Service Operation and ID

Operation	ID
FOLDER_CREATE_SHORTCUTS	20072
FOLDER_DELETE_SHORTCUTS	20073
FOLDER_GET_DOCUMENT_ID_BY_PATH	20074
FOLDER_GET_FOLDER_ID_BY_PATH	20075
FOLDER_GET_OBJECT_BY_PATH	20076
FOLDER_GET_PATH_FOR_OBJECT	20077
DRAWER_WITH_CREATE_PRIV_GET_LIST	20078
FOLDER_MOVE_OBJECT_BY_PATH	20079
IN_VIEW_GET_ALL	20080
DOCUMENT_ADD_TO_FOLDER	20081
DOCUMENT_REMOVE_FROM_FOLDER	20082

Folder Service Error Codes and Messages

Error Code	Error Message
6000	Message Agent could not create the folder.
6001	Message Agent could not delete the folder.
6002	Message Agent could not get the folder.
6003	Message Agent could not get all the folders.
6004	Message Agent could not get the folder documents.
6005	Message Agent could not get the folder type.
6006	Message Agent could not get all folder types.

Error Code	Error Message
6007	Message Agent could not update the folder.

Form Web Service

The following tables provide the IDs, error codes, and error messages that can be returned as exceptions for operation in the Form Web service.

Form Service Operation and ID

Operation	ID
FORM_ADD_SHARED_FILE	41001
FORM_ADD_DATA_DEFINITION	41002
FORM_ADD_PRESENTATION	41003
FORM_UPDATE_PRESENTATION	41004
INVALID_FILE_TYPE	41005
FORM_GET_ALL_SHARED_FILE	41006
FORM_GET_SHARED_FILE	41007
FORM_GET_ALL_DATA_DEFINITION	41008
FORM_GET_DATA_DEFINITION	41009
FORM_GET_ALL_PRESENTATION	41010
FORM_GET_PRESENTATION	41011
FORM_GET_ALL	41012
FORM_GET	41013
FORM_ADD	41014
FORM_UPDATE	41015
FORM_PUBLISH_BY_ID	41016
FORM_PUBLISH_BY_FILE	41017

Form Service Error Codes and Messages

Error Code	Error Message
8025	The name is invalid.
8026	Must remove using form presentation file by ID.
8027	Must replace using form presentation by target ID and source file.
8028	ID is a required parameter.
8029	File is empty.
8030	Message Agent failed to get physical form file.
8032	The ID length is invalid.

Task Web Service

The following tables provide the IDs, error codes, and error messages that can be returned as exceptions for operations in the Task Web service.

Task Service Operation IDs

Operation	ID
TASK_GET_PROPERTIES	40000
MY_ASSIGNED_TASKS	40001
TASK_ADD_COMMENT	40002
TASK_CREATE	40003
TASK_COMPLETE	40004
TASK_GET_ALL_REASONS	40005

Task Service Error Codes and Messages

Error Code	Error Message
4000	The user cannot retrieve the requested task information. Contact your ImageNow administrator for assistance.
4001	Requested task template not found.
4002	No action or return reasons exist for the requested task template ID.

Workflow Web Service

The following tables provide the IDs, error codes, and error messages that can be returned as exceptions for operations in the Workflow Web service.

Workflow Service Operation IDs

Operation	ID
WORKFLOW_QITEM_SET	30000
WORKFLOW_QITEM_GET	30001
WORKFLOW_QITEM_DELETE	30002
WORKFLOW_QITEM_GET_LIST	30003
WORKFLOW_QITEM_ROUTE_AUTO	30004
WORKFLOW_QITEM_ROUTE_MANUAL	30005
WORKFLOW_QITEM_ROUTE_BACK	30006
WORKFLOW_QITEM_SET_STATUS	30007
WORKFLOW_QITEM_GET_HISTORY	30008
WORKFLOW_Q_GET_LIST	30009
WORKFLOW_Q_GET_USERS	30010
WORKFLOW_DOC_GET_INFO	30011
WORKFLOW_QITEM_GET_STATUS	30012
WORKFLOW_QITEM_NOTIFY_NEW	30013
WORKFLOW_QITEM_GET_NEXT	30014
WORKFLOW_QITEM_GET_INFO	30015
WORKFLOW_GET_USER_DOC_INFO	30016
WORKFLOW_PROCESS_GET_LIST	30018
WORKFLOW_QITEM_SET_HOLD	30019
WORKFLOW_Q_GET_HOLD_REASON_LIST	30020
WORKFLOW_Q_GET_ROUTES_FORWARD	30021

Workflow Service Error Codes and Messages

Error Code	Error Message
3001	Queue name must not be empty.
3002	Item ID is invalid or not in the correct format.

General Error Messages

Message Agent also returns some general faults which can apply to any operation. When an operation fails, the exception returns an error code and error message from the following table.

General Error Codes and Messages

Error Code	Error Message
8000	The operation failed.
8001	Message Agent could not get Unique ID.
8002	Message Agent could not get User ID.
8003	Message Agent could not get current time.
8004	Message Agent could not unlock all of the documents for this user.
8005	This document is not in the workflow.
8006	The document ID is either invalid or not in the correct format.
8007	The drawer or document type is empty.
8008	Message Agent could not create the directory.
8009	The record number returned is incorrect.
8010	Failed to switch user.
8011	Failed to find the user session.
8012	The connection was lost.
8013	Message Agent failed to get license information.
8014	The user is not a valid bridge.
8015	This user doesn't have proper privileges.
8016	Message Agent failed to get privileges.
8017	Message Agent failed to get transaction license.
8018	The specified user does not have a document open in workflow.
8019	The required ImageNow user name value was not provided.
8020	Document keys are invalid.
8021	The project is invalid.
8022	Message Agent did not find a workflow item in the queue.
8023	Nothing found.
8024	Multiple documents match the provided document key values.
8888	The user has not been authenticated.

Index

about WSDL and XSD files	8	overview of	
access service URL	10	the access service.....	7
add a folder associated with a document	14	the document service.....	7
audit a user	11	the folder service.....	7
authentication/authorization	10	the form service	7
bridge user	11	the task service	7
delete a workflow item	17	the workflow service	8
details on a specific workflow item.....	16	retrieve assigned tasks	16
document service URL	10	return a document object	14
documentation		return an array of information for a form	15
help.....	7	service operations	
XML	7	imagenow_access.wsdl	10
end a user session	11	imagenow_document.wdsl	13
error IDs and codes	17	imagenow_folder.wsdl	14
access service codes	18	imagenow_form.wsdl	15
document service codes	18	imagenow_task.wsdl.....	15
folder service codes	21	imagenow_workflow.wsdl.....	16
general codes	25	SESSION_STRING parameter	12
task service codes.....	23	set a document to be managed in workflow ...	16
workflow service codes.....	24	SOAP	5, 6
folder service URL.....	10	SOAP/XML.....	6
form service URL	10	standards	
generating client proxy	9	WC3 - World Wide Web Consortium	6
get information for a specific document.....	13	WS-I – Web Services Interoperability	6
hierarchical folder path	14	task service URL.....	10
HTTP / HTTPS	5, 6	TCP/IP.....	5, 6
identify form attributes	15	update folder attributes	14
imagenow_commonTypes.xsd	8, 9	workflow service URL.....	10
INOW_BOOLEAN_RESPONSE element.....	11	WSDL files	5, 6, 8
INOW_CONTEXT element.....	12	XML.....	5, 6
log in as an individual user.....	11	XML Schema	8
login methods	11	XSD files.....	9