GENESYS

AN ALCATEL-LUCENT COMPANY

intelligent Workload Distribution 7.6

JMS Capture Adapter

Reference Manual

About Genesys

Alcatel-Lucent's Genesys solutions feature leading software that manages customer interactions over phone, Web, and mobile devices. The Genesys software suite handles customer conversations across multiple channels and resources—self-service, assisted-service, and proactive outreach—fulfilling customer requests and optimizing customer care goals while efficiently using resources. Genesys software directs more than 100 million customer interactions every day for 4,000 companies and government agencies in 80 countries. These companies and agencies leverage their entire organization, from the contact center to the back office, while dynamically engaging their customers. Go to www.genesyslab.com for more information.

Each product has its own documentation for online viewing at the Genesys Technical Support website or on the Documentation Library DVD, which is available from Genesys upon request. For more information, contact your sales representative.

Notice

Although reasonable effort is made to ensure that the information in this document is complete and accurate at the time of release, Genesys Telecommunications Laboratories, Inc., cannot assume responsibility for any existing errors. Changes and/or corrections to the information contained in this document may be incorporated in future versions.

Your Responsibility for Your System’s Security

You are responsible for the security of your system. Product administration to prevent unauthorized use is your responsibility. Your system administrator should read all documents provided with this product to fully understand the features available that reduce your risk of incurring charges for unlicensed use of Genesys products.

Trademarks

Technical Support from VARs

If you have purchased support from a value-added reseller (VAR), please contact the VAR for technical support.

Technical Support from Genesys

If you have purchased support directly from Genesys, please contact Genesys Technical Support at the following regional numbers:

Region	Telephone	E-Mail
North and Latin America	+888-369-5555 or +506-674-6767	support@genesyslab.com
Europe, Middle East, and Africa	+44-(0)-1276-45-7002	support@genesyslab.co.uk
Asia Pacific	+61-7-3368-6868	support@genesyslab.com.au
Japan	+81-3-6361-8950	support@genesyslab.co.jp

Prior to contacting technical support, please refer to the *Genesys Technical Support Guide* for complete contact information and procedures.

Ordering and Licensing Information

Complete information on ordering and licensing Genesys products can be found in the ***Genesys Licensing Guide*****.**

Released by

Genesys Telecommunications Laboratories, Inc. www.genesyslab.com

Document Version: 76iwd_ref_jms-ca_08-2010_v7.6.101.00

Table of Contents

List of Procedures

Preface

Welcome to the *intelligent Workload Distribution 7.6 JMS Capture Adapter Reference Manual*. This document provides a detailed description of the intelligent Workload Distribution (iWD) Java Message Service (JMS) interface, to guide you in implementing iWD integration with the JMS Capture Point.

This document is valid only for the 7.6 release of this product.

Note: For versions of this document created for other releases of this product, please visit the Genesys Technical Support website, or request the Documentation Library DVD, which you can order by e-mail from Genesys Order Management at orderman@genesyslab.com.

This preface provides an overview of this document, identifies the primary audience, introduces document conventions, and lists related reference information:

The iWD JMS Capture Adapter is a component that allows iWD to capture tasks from source systems that use Java Message Service (JMS) as a message bus.

Intended Audience

This document is intended for architects and developers who want to implement a project that leverages iWD via the JMS interface. It has been written with the assumption that you have a basic understanding of:

- intelligent Workload Distribution (iWD) and JMS messaging services.
- Network design and operation

- Your own network configurations

Chapter Summaries

In addition to this preface, this document contains the following chapters :

- Chapter 1, "iWD JMS Capture Adapter Overview," on page 13, provides an overview of the JMS Capture Adapter, how it is installed, and how to configure the corresponding JMS Capture Point.
- Chapter 2, "iWD Messages," on page 27, describes the XML schema used for messaging.

Document Conventions

This document uses certain stylistic and typographical conventions—introduced here—that serve as shorthands for particular kinds of information.

Document Version Number

A version number appears at the bottom of the inside front cover of this document. Version numbers change as new information is added to this document. Here is a sample version number:

76fiwd_jms-ca_07-2010_v7.6.101.00

You will need this number when you are talking with Genesys Technical Support about this product.

Type Styles

Italic

In this document, italic is used for emphasis, for documents' titles, for definitions of (or first references to) unfamiliar terms, and for mathematical variables.

Examples:

- Please consult the *Genesys Migration Guide* for more information.
- *A customary and usual practice* is one that is widely accepted and used within a particular industry or profession.
- Do *not* use this value for this option.
- The formula, $x + 1 = 7$ where x stands for . . .

Monospace Font

A monospace font, which looks like `teletype or typewriter text`, is used for all programming identifiers and GUI elements.

This convention includes the *names* of directories, files, folders, configuration objects, paths, scripts, dialog boxes, options, fields, text and list boxes, operational modes, all buttons (including radio buttons), check boxes, commands, tabs, CTI events, and error messages; the values of options; logical arguments and command syntax; and code samples.

Examples:

- Select the `Show variables on screen` check box.
- Click the `Summation` button.
- In the `Properties` dialog box, enter the value for the host server in your environment.
- In the `Operand` text box, enter your formula.
- Click `OK` to exit the `Properties` dialog box.
- The following table presents the complete set of error messages T-Server distributes in `EventError` events.
- If you select `true` for the `inbound-bsns-calls` option, all established inbound calls on a local agent are considered business calls.

Monospace is also used for any text that users must manually enter during a configuration or installation procedure, or on a command line:

Example:

- Enter `exit` on the command line.

Screen Captures Used in This Document

Screen captures from the product GUI (graphical user interface), as used in this document, may sometimes contain a minor spelling, capitalization, or grammatical error. The text accompanying and explaining the screen captures corrects such errors *except* when such a correction would prevent you from installing, configuring, or successfully using the product. For example, if the name of an option contains a usage error, the name would be presented exactly as it appears in the product GUI; the error would not be corrected in any accompanying text.

Square Brackets

Square brackets indicate that a particular parameter or value is optional within a logical argument, a command, or some programming syntax. That is, the parameter’s or value’s presence is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information. Here is a sample:

```
smcp_server -host [/flags]
```

Angle Brackets

Angle brackets indicate a placeholder for a value that the user must specify. This might be a DN or port number specific to your enterprise. Here is a sample:

```
smcp_server -host <confighost>
```

Related Resources

Consult these additional resources as necessary:

- *intelligent Workload Distribution 7.6 Overview*, which provides an overview of the product.
- *intelligent Workload Distribution 7.6 Deployment Guide*, which describes deployment procedures for all intelligent Workload Distribution components.
- *intelligent Workload Distribution 7.6 High Availability Guide*, which describes ways of providing a high degree of system availability for iWD.
- *intelligent Workload Distribution 7.6 Manager Guide*, which describes the use of iWD for task monitoring and task operations.
- *intelligent Workload Distribution 7.6 MQ Capture Point API Reference Guide*, which provides a detailed description of the iWD MQ interface and guides you in implementing iWD integration via the MQ Capture Point.
- *intelligent Workload Distribution 7.6 Data Mart Reference Guide*, which describes the data schemas that compose the iWD Data Mart to guide you in the design of reports based on iWD data.
- The *Genesys Technical Publications Glossary,* which ships on the Genesys Documentation Library DVD and which provides a comprehensive list of the Genesys and CTI terminology and acronyms used in this document.
- The *Genesys Migration Guide*, also on the Genesys Documentation Library VDD, which provides a documented migration strategy from Genesys product releases 6.x and later to all Genesys 7.x releases. Contact Genesys Technical Support for additional information.
- The Release Notes and Product Advisories for this product, which are available on the Genesys Technical Support website at http://genesyslab.com/support.

Genesys product documentation is available on the:

- Genesys Technical Support website at http://genesyslab.com/support.
- Genesys Documentation Library DVD, which you can order by e-mail from Genesys Order Management at orderman@genesyslab.com.

Making Comments on This Document

If you especially like or dislike anything about this document, please feel free to e-mail your comments to Techpubs.webadmin@genesyslab.com.

You can comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this document. Please limit your comments to the information in this document only and to the way in which the information is presented. Speak to Genesys Technical Support if you have suggestions about the product itself.

When you send us comments, you grant Genesys a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

Document Change History

This is the first release of the iWD 7.6 JMS Capture Adapter Reference Manual. In the future, this section will list topics that are new or have changed significantly since the first release of this document.

Chapter

1 iWD JMS Capture Adapter Overview

This chapter explains the intelligent Workload Distribution (iWD) Java Message Service (JMS) Capture Adapter architecture and the components that are involved. The information in this chapter is organized into the following topics:

What is the iWD JMS Capture Adapter?

The iWD JMS Capture Adapter is a capture adapter that allows iWD to capture tasks from source systems that use Java Message Service (JMS) as a message bus. The Capture Adapter provides a fully bidirectional link and supports the full iWD API (such as task creation, updating, holding, canceling, and various task state change notifications). The iWD 7.6.1 JMS Capture Adapter is supported only with the 7.6.1 release of iWD.

Note: The JMS Capture Adapter supports the *point-to-point* domain, as described in the JMS 1.1 specification.

iWD JMS Capture Adapter Architecture

The iWD JMS Capture Adapter uses JMS version 1.1 API to connect to any J2EE-compliant JMS Provider. In iWD configuration, a JMS capture point is set up, which is a specific configuration object. The Capture Adapter is a technology solution that provides the actual integration point to the JMS provider, whereas the capture point is a configuration object against which business rules will be defined. Figure 1 illustrates the architecture.

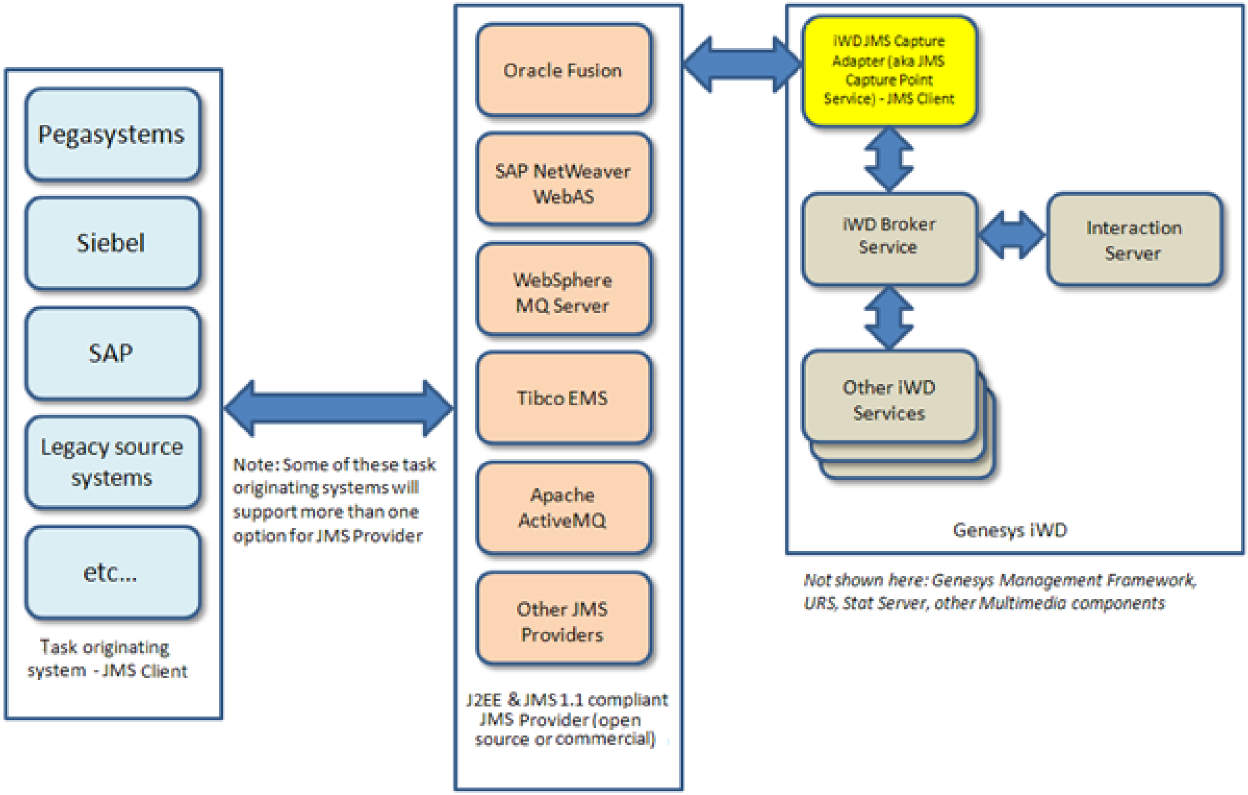

Pegasystems
Siebel
SAP
Legacy source systems
etc...
Task originating system - JMS Client
Note: Some of these task originating systems will support more than one option for JMS Provider
Oracle Fusion
SAP NetWeaver WebAS
WebSphere MQ Server
Tibco EMS
Apache ActiveMQ
Other JMS Providers
J2EE & JMS 1.1 compliant JMS Provider (open source or commercial)
iWD JMS Capture Adapter (aka JMS Capture Point Service) - JMS Client
iWD Broker Service
Interaction Server
Other iWD Services
Genesys iWD
Not shown here: Genesys Management Framework, URS, Stat Server, other Multimedia components

Figure 1: JMS Capture Adapter Architecture

The JMS Capture Adapter supports connectivity to any customer-supplied J2EE-compliant JMS Provider and utilizes two queues: input (`GTLIn`) and output (`GTLOut`). The names of the queues are configurable, using the `queueInbound` and `queueOutbound` properties (described in Table 1 on page 20).

- The input queue is used to receive messages from the task originating system, such as task creation or task update requests.
- The output queue is used to send back responses to requests, as well as to send notifications about task state changes that are not triggered by the originating system (such as when a task gets assigned to an agent).

Note: The JMS Capture Adapter, like all others in iWD, supports multiple Capture Points.

The JMS Capture Adapter also supports optional message transformation. In this case, each incoming XML message and each outgoing message is passed through transformation scripts, thus allowing integration with custom XML formats. Figure 2 depicts the transformation process.

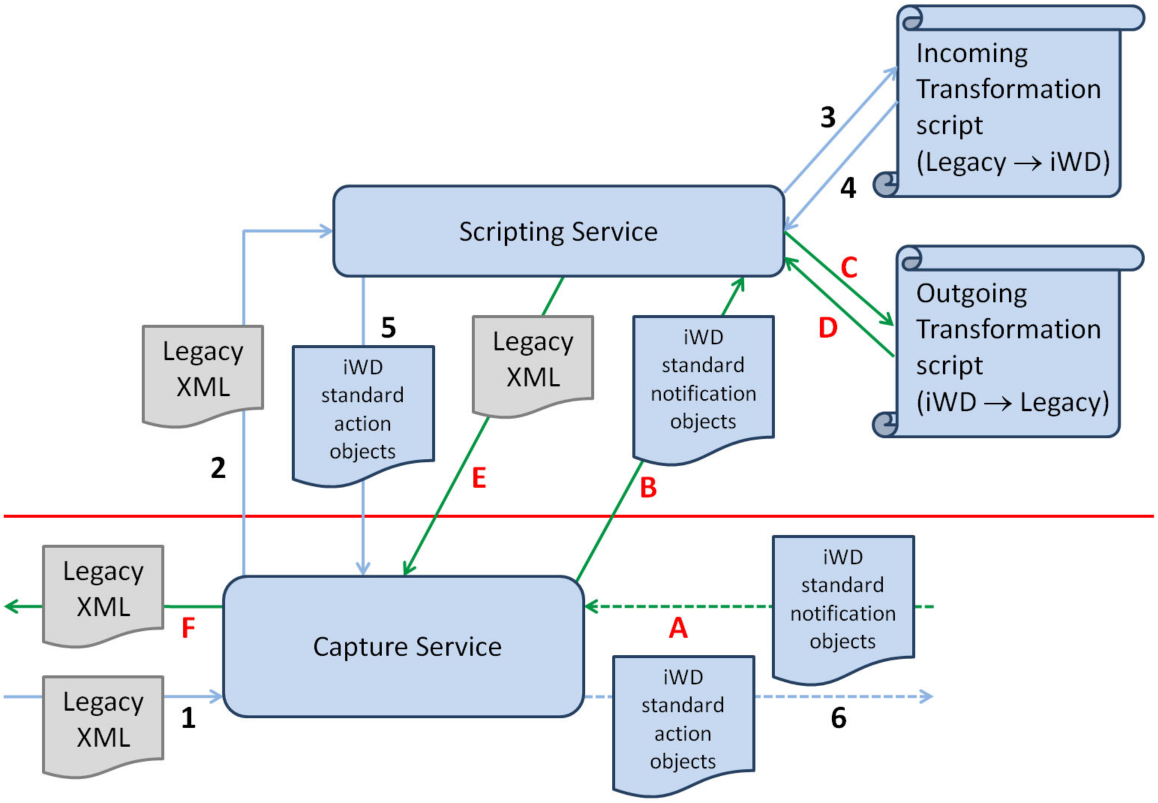

Incoming Transformation script (Legacy → iWD)
Outgoing Transformation script (iWD → Legacy)
Scripting Service
Capture Service
Legacy XML
iWD standard action objects
iWD standard notification objects
1
2
3
4
5
6
A
B
C
D
E
F

Figure 2: Transformation

The numbers in Figure 2 identify the steps in the flow when transformation is used:

1. The Legacy XML is captured.
2. The Legacy XML is passed to the Scripting Service.
3. The Scripting Service executes the Incoming Transformation script, using the Legacy XML file as input.
4. The Incoming Transformation script produces iWD standard action objects as output, which are passed to the Scripting Service.
5. The Scripting Service passes the iWD standard action objects to the Capture Service.
6. The task is created. (The dotted lines indicate communication with other iWD components).

The letters identify the steps in the flow for the reverse process:

A. The Capture Service receives notification about a task.

B. The Capture Service gets task notification objects and passes them to the Scripting Service.

C. The Scripting Service executes the Outgoing Transformation script, using the iWD standard notification objects as input.

D. The Outgoing Transformation script produces the Legacy XML as output and passes it to the Scripting Service.

E. The Scripting Service passes the newly created Legacy format XML to the Capture Service.

F. The Capture Service writes the Legacy format XML to an MQ or XML file to be used to respond back to the legacy system about requests or task state change notifications (for example, if the task has been assigned to an agent).

If transformation is not used, the steps above the red line in Figure 2 are not invoked. In this case, the incoming messages from the legacy system must be in iWD standard XML format. This document describes the standard iWD XML message formats that are supported without the transformation. Each message is formatted as an XML string in which the root node is always named `GTLMessages`. The child nodes of `GTLMessages` indicate the message type, such as `CreateTask`.

The types and formats of the input and output messages that are supported are described in Chapter 2, "iWD Messages," on page 27.

iWD JMS Capture Point Configuration

The JMS Capture Point is an iWD service for capturing and manipulating tasks via JMS (Java Messaging System). The messages must be in XML format.

iWD provides a standard iWD XML message schema (described in detail in Chapter 2, "iWD Messages," on page 27). Alternatively, it is possible to use custom XML formats via message transformation scripts.

To handle custom XML formats, transformation scripts must be created for each direction needed: typically one for input transformation, and one for output. iWD currently supports the Groovy scripting language, as well as JavaScript, for these purposes. For more information on transformation scripts, see the standard iWD input and output transformation scripts that are included in the iWD core package.

This section describes how to deploy the JMS Capture Adapter and configure its capture point service.

Deploying the JMS Capture Adapter

Before installing the adapter, you need to ensure your environment meets the following requirements and prerequisites:

- The iWD JMS Capture Adapter is supported on the same platforms as the iWD 7.6.1 Solution. Refer to the *Genesys Supported Operating Environment Reference Manual* for information.

- For Tomcat:
  - Ensure your JMS provider's jar files are in the search path.
- For WebSphere:
  - Configure required parameters in the WebSphere Admin Console the JMS provider. This includes the JNDI names for the JMS connection factory and the JMS input and output queues.

Procedure: Installing the JMS Capture Adapter

Purpose: To install the JMS Capture Adapter.

Start of procedure

1. On the JMS Capture Adapter CD, locate and double-click `setup.exe`.
2. Click `Next` on the welcome screen.
3. Select the destination directory for the JMS Capture Adapter installation. The deployment files, a test program, and sample test input are copied to this location. You can select the default, or standard, Genesys location (on Windows, this would be `C:\Program Files\GCTI\iWD JMS Capture Adapter`) or browse to another location. Click `Next`.
4. Click `Install`, and then click `Next` when installation is complete.
5. Locate the deployment .jar file (`evo.gtl.capture.jms.jar`) in the directory specified in Step 3 and move it to the correct location on your application server and/or execute the proper steps to deploy it. This .jar file needs to be added to the iWD Management node as well as iWD runtime nodes.

End of procedure

Next Steps

- Configure the JMS Capture Point, as described in "Creating the JMS Capture Point Service".

Creating the JMS Capture Point Service

The JMS Capture Point Service template is not imported into the iWD configuration database by default; it has to be imported manually. This section describes how to import the service template and configure the service.

Procedure: Importing the JMS Capture Point service template

Purpose: To import the JMS Capture Point service template and assign it as a Module for your Solution.

Start of procedure

1. Log into iWD Manager.

> **Note:** For a detailed description of the iWD Manager interface, including logging in, the interface layout, and available functionality, refer to the *iWD 7.6 Manager Guide* and *iWD Manager Help*.

2. Select the `System` tenant from the tenant selection drop-down list.
3. In the navigation tree, select `Import/Export`.
4. On the Import/Export details view, click `Browse` and navigate to the installation directory for the JMS Capture Adapter, as mentioned in Step 3 of Procedure: Installing the JMS Capture Adapter, on page 17.
5. From the `config` directory select the `gtl_jms.xml` file to import.
6. Click `Import`. An `Import Successful` confirmation message will appear.
7. Select your tenant from the list of managed tenants.
8. Move the iWD JMS Capture Service to the list of Assigned Modules. Click `Save`.
9. Select the tenant from the tenant selection drop-down list.
10. Select the `Services` navigation section, and then select your Solution in the navigation tree.
11. In the Solution Details view, move the iWD JMS Capture Point service to the list of Assigned Modules. Click `Save`.
12. The iWD JMS Capture Point is now available under the Service templates for your Solution.

End of procedure

Next Steps

- Configure the JMS Capture Point service. See Procedure: Configuring the JMS Capture Point service.

Procedure: Configuring the JMS Capture Point service

Purpose: To create the service in iWD Manager based on the template, and to configure the properties for the service.

Start of procedure

1. In iWD Manager, select the Services navigation section.
2. Locate your Solution in the navigation tree. Expand the Services node in the navigation tree (if necessary), and click `New Service`.
3. Select the iWD JMS Capture Point service from the templates drop down list, and configure the remaining properties as described in Table 1 on page 20.
4. On the Service details screen, ensure the correct iWD Runtime Node is selected if your environment has multiple runtime nodes. This ensures a more scalable solution by allowing you to distribute the iWD services across multiple nodes, and is also used for high availability configurations where you would want primary and backup JMS Capture Point services to be running on separate runtime nodes.
5. When configuration is complete, click `Save`. Remember all changes have to be deployed before they take effect.

End of procedure

Next Steps

- A test program is included in the installation of the JMS Capture Adapter. Refer to the `readme.txt` file found in the test folder of the installation directory (specified in Step 3 on page 17) for more information about the test utility.

Table 1 lists the configurable properties for the JMS Capture Point service.

Table 1: JMS Capture Point Service properties

Property	Description
startAutomatically	Indicates whether the service should be started automatically after the configuration deployment.
logLevel	The Service log level. This should be set to `Default` unless otherwise instructed by Genesys Technical Support. The log levels are: • `Default`—the service will use the logging level configured in the Logging Service. • `Debug`—the most detailed informational events that are most useful in debugging an application. • `Info`—informational messages that highlight the progress of the application. • `Warn`—potentially harmful situations. • `Error`—error events that might not affect the application's ability to run. • `Fatal`—severe error events. • `All`—turns on all logging. • `Off`—turns off all logging.
logAgeInDays	Sets the number of days that log files should be kept in the system. A value of `0` disables this limit. This property is not available if `logLevel` is set to `Off` or `Default`.
logMaxFileSizeMegaBytes	Sets a limit on the size of a single log file, in megabytes. A value of `0` disables this limit. This property is not available if `logLevel` is set to `Off` or `Default`.
logFilesToKeep	Sets a limit on the number of log files that are kept for this service, excluding the current log file. A value of `0` disables this limit. This property is not available if `logLevel` is set to `Off` or `Default`.
checkIfAlreadyCaptured	If `true`, iWD will verify whether another task that has a given `captureId` has already been captured.
timeZone	The time zone of the JMS Capture Point. Date and time values will be converted from the specified time zone to UTC before those values are stored in iWD. In addition, any date and time values that are included in response messages will be converted to the specified time zone. If this parameter is not specified, it defaults to the tenant time zone.

Table 1: JMS Capture Point Service properties (Continued)

Property	Description
BrokerService	Mandatory dependency: The Broker Service used by the JMS Capture Point. For more information on the Broker Service, refer to the *iWD Deployment Guide*.
addPrefixToJNDINames	If selected, the prefix `java:comp/env` is added to the JNDI (Java Naming and Directory Interface) names for the connection factory and the JMS queues. The default value for this property is cleared (false). For WebSphere, leave this property cleared. For Tomcat, select this property only if the `setJNDINamingContext` option is **not** selected.
connectionFactoryJNDI	The JNDI name of a JMS `ConnectionFactory`. A `ConnectionFactory` object encapsulates a set of JMS connection configuration parameters that has been defined by an administrator. The JMS Capture Adapter uses it to create a connection with the JMS provider. The JMS API establishes the convention that JMS clients find administered objects by looking them up in a JNDI namespace. See “ConnectionFactory and Queue Sample Definitions” on page 25.
userName	An optional username for authentication when creating a connection through the connection factory. If not specified, the default identity defined in the connection factory is used (if one has been defined).
password	An optional password for authentication when creating a connection through the connection factory. The password is displayed in asterisks (*) as it is typed, and stored encrypted in the configuration.
queueInboundJNDI	The JNDI name of the inbound JMS `Queue`. A `Queue` object encapsulates a provider-specific queue name. This is how a client specifies the identity of a queue to JMS API methods. Requests to the Capture Point from the task originating system are read from this queue.
queueOutboundJNDI	The JNDI name of the outbound JMS `Queue`. Notifications from the Capture Point to the task originating system are written to this queue.

Table 1: JMS Capture Point Service properties (Continued)

Property	Description
queueWorkerThreads	The number of concurrent message consumers to create. The default is 1. These consumers receive messages from the inbound JMS Queue. Specifying a higher value for this property will increase the standard level of scheduled concurrent consumers at runtime. **Note:** If this thread count is greater than 1, then the order in which inbound requests are processed may not be the same order in which they are queued.
reconnectTimeoutMilliseconds	The interval between connection recovery attempts, in milliseconds.
setJNDINamingContext	Select this option to configure (within the JMS Adapter) the JNDI context for your JMS provider. Checking this option displays the additional properties. Do **not** check this option and configure the additional properties if: • You use WebSphere. For WebSphere, the JMS context is defined through the WebSphere Management Console, not in iWD Manager. • You use Tomcat, and have **already** defined your JNDI context in a `context.xml` file.
initialContextFactory	The Java class name of the context factory supplied by your JMS provider. This class, and associated classes, must be in your Java path. This property appears only if `setJNDINamingContext` is selected.
providerURL	The URL of your JMS provider. This property appears only if `setJNDINamingContext` is selected.
contextKey<x>	In setting the JNDI context, each JMS provider may have provider-specific parameters. Up to 5 key-value parameters can be entered (for example, `contextKey1`, `contextKey2`, and so on). These key-value pairs are passed to the initial context factory. This property appears only if `setJNDINamingContext` is selected.

Table 1: JMS Capture Point Service properties (Continued)

Property	Description
contextValue<x>	The corresponding value for the key specified in `contextKey<x>`. This property appears only if `setJNDINamingContext` is selected.
TransformScriptingService	Optional dependency: Scripting Service (see the *iWD Deployment Guide* for more information about the Scripting Service). If specified, XML input/output will be transformed using input/output transformation scripts.
TransformInputScript	The input transformation script; must be specified when `TransformScriptingService` is selected.
TransformOutputScript	The output transformation script; must be specified when `TransformScriptingService` is selected.
TransformOutputRootNode	The name of the XML root node in output files. This property is visible only when `TransformScriptingService` is selected.
NotifyError	If selected, indicates that error notifications will be sent to an outbound queue; available only when transformation is not enabled.
NotifyTaskAssigned	If selected, indicates that “task assigned” notifications will be submitted to an outbound queue; available only when transformation is not enabled.
NotifyTaskCanceled	If selected, indicates that “task cancelled” notifications will be submitted to an outbound queue; available only when transformation is not enabled.
NotifyTaskCreated	If selected, indicates that “task created” notifications will be submitted to an outbound queue; available only when transformation is not enabled.
NotifyTaskRejected	If selected, indicates that “task rejected” notifications will be submitted to an outbound queue; available only when transformation is not enabled.
NotifyTaskCompleted	If selected, indicates that “task completed” notifications will be submitted to an outbound queue; available only when transformation is not enabled.
NotifyTaskDistributed	If selected, indicates that “task distributed” notifications will be submitted to the outbound queue; available only when transformation is not enabled.

Table 1: JMS Capture Point Service properties (Continued)

Property	Description
NotifyTaskDistributedQueue	If selected, indicates that “task moved into distribution queue” notifications will be submitted to an outbound queue; available only when transformation is not enabled.
NotifyTaskFinished	If selected, indicates that “task finished” notifications will be submitted to an outbound queue; available only when transformation is not enabled.
NotifyTaskHeld	If selected, indicates that “task held” notifications will be submitted to an outbound queue; available only when transformation is not enabled.
NotifyTaskRestarted	If selected, indicates that “task restarted” notifications will be submitted to an outbound queue; available only when transformation is not enabled.
NotifyTaskReturned	If selected, indicates that “task returned” notifications will be submitted to an outbound queue; available only when transformation is not enabled.
NotifyTaskResumed	If selected, indicates that “task resumed” notifications will be submitted to an outbound queue; available only when transformation is not enabled.
NotifyTaskUpdated	If selected, indicates that “task updated” notifications will be submitted to an outbound queue; available only when transformation is not enabled.
NotifyTaskUpdatedOptions	This property is only available when the `NotifyTaskUpdated` property is selected. The possible values are: • `Always`—always notify the task origination system when a task has been updated. • `Internal_Update_Only`—only notify the task origination system when a task is updated internally by iWD. • `Requested_Update_Only`—only notify the task origination system when a task is updated by a request from the task origination system.
backupFor	High availability: The primary JMS Capture Point in a high availability setup scenario. This service will perform a backup role in such a setup, and will take over processing if the primary service becomes unavailable. For more information about High Availability, refer to the *iWD High Availability Guide.*

ConnectionFactory and Queue Sample Definitions

The following is a sample definition of an Active MQ `ConnectionFactory` object and two `Queue` objects (as defined for Tomcat in the `context.xml` file):

Note: This section contains *sample* configurations for possible JMS Providers. However, the JMS Capture Adapter is based on the JMS 1.1 standard, which means it will work with *any* JMS Provider that supports that standard. Refer to the documentation for your specific JMS Provider to determine the proper configuration required on your provider.

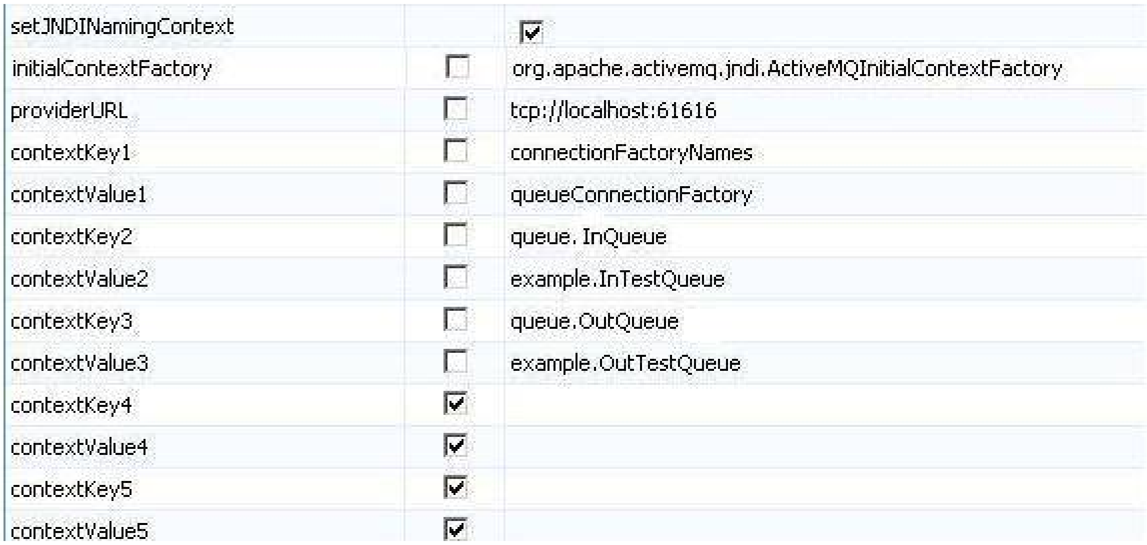
```
<Resource name="queueConnectionFactory" auth="Container"
              type="org.apache.activemq.ActiveMQConnectionFactory"
              description="JMS Connection Factory"
              factory="org.apache.activemq.jndi.JNDIReferenceFactory"
              brokerURL="tcp://localhost:61616"
              useEmbeddedBroker="false" />

<Resource name="InQueue" auth="Container"
              type="org.apache.activemq.command.ActiveMQQueue"
              description="JMS inbound Queue"
              factory="org.apache.activemq.jndi.JNDIReferenceFactory"
              physicalName="example.InTestQueue" />

<Resource name="OutQueue" auth="Container"
              type="org.apache.activemq.command.ActiveMQQueue"
              description="JMS outbound Queue"
              factory="org.apache.activemq.jndi.JNDIReferenceFactory"
              physicalName="example.OutTestQueue" />
```

When using Tomcat there are two ways in which the JNDI context may be configured: in the `context.xml` file as the example above shows, **or** with the options activated by selecting the `setJNDINamingContext` option (see page 22 for option descriptions). Genesys recommends that you use the latter method, using the the options in iWD Manager. You do **not** have to configure in both iWD Manager and the `context.xml` file.

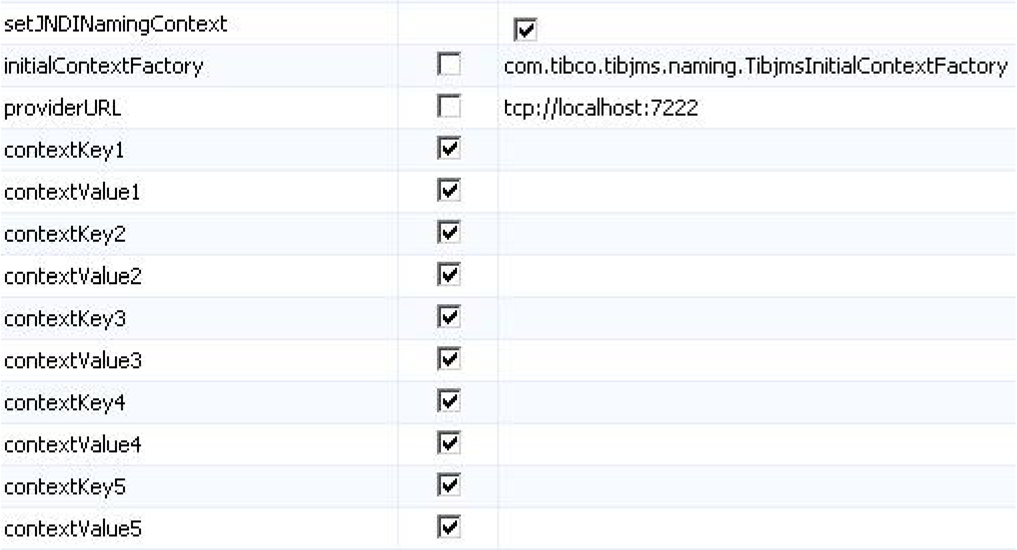
Figure 3 shows an example of how an ActiveMQ context factory might be configured in the JMS Capture Point service in iWD Manager using the configuration options.



setJNDINamingContext		☑
initialContextFactory	□	org.apache.activemq.jndi.ActiveMQInitialContextFactory
providerURL	□	tcp://localhost:61616
contextKey1	□	connectionFactoryNames
contextValue1	□	queueConnectionFactory
contextKey2	□	queue. InQueue
contextValue2	□	example.InTestQueue
contextKey3	□	queue.OutQueue
contextValue3	□	example.OutTestQueue
contextKey4	☑	
contextValue4	☑	
contextKey5	☑	
contextValue5	☑	

Figure 3: Example ActiveMQ Configuration

Figure 4 shows an example of how a Tibco EMS context factory might be configured.



setJNDINamingContext		☑
initialContextFactory	□	com.tibco.tibjms.naming.TibjmsInitialContextFactory
providerURL	□	tcp://localhost:7222
contextKey1	☑	
contextValue1	☑	
contextKey2	☑	
contextValue2	☑	
contextKey3	☑	
contextValue3	☑	
contextKey4	☑	
contextValue4	☑	
contextKey5	☑	
contextValue5	☑	

Figure 4: Example Tibco EMS Configuration

Connectivity to JMS Providers Over TLS

Connectivity to JMS Providers over TLS (Transport Layer Security) is supported by the JMS Capture Adapter if it is supported by the JMS Provider. For the use of TLS, any additional parameters required are configured in the ConnectionFactory resource definition. Refer to the documentation for your JMS Provider for more information.

Chapter

2 iWD Messages

This chapter provides a detailed description of all of the input and output iWD Messages that are supported by the capture adapter. The Unicode character set is supported within messages but the encoding **must** be of type UTF-8.

The information in this chapter is organized into the following topics:

Data Types

Table 2 describes the data types used in iWD messages.

Table 2: Data Types for iWD Messages

Type	Description
Integer	An integer value ($–2^{31}$ <value <2^{31}).
String	A string value. The maximum length is specified in parentheses, where applicable.
Boolean	A Boolean value (`true` or `false`).
DateTime	A date/time value. Date/time should be formatted according to the ISO 8601 standard `YYYY-MM-DDThh:mm:ss` (for example 2007-08-26T21:32:00) and should be provided for the time zone that is configured for the given Capture Point.

Messages

The following information is documented for each message:

- Direction—“In” or “Out”. Every “In” message comes from the originating system and results in an “Out” message, unless the specific “Out” message is turned off in capture point configuration. “Out” messages are sent as responses to “In” messages, as well as notifications when the task state changes within the iWD.
- Description—A functional description of the message.
- Format—The XML format of the message. This illustrates a structure of the entire XML message, by using data types (see Table 2 on page 27) instead of node values.
- Attributes—A description of each attribute that is used in the XML message.
- Response messages—The response messages that this message can trigger. Response messages are applicable only for messages that have an “In” direction.
- Error codes—The error codes that this message can return by using the “Error” message. Error codes are applicable only for messages that have an “In” direction.

Task Action

Direction: In

Description: This section describes common attributes and responses for all inbound messages.

Format:

```
<GTLMessages>
   <[action name]>
      <BrokerId>Integer</BrokerId> or <CaptureId>String(64)</CaptureId>
      <Actor>String(255)</Actor>
      <Reason>String(255)</Reason>
      <ActionDateTime>DateTime</ActionDateTime>
      [attributes specific to action]
   </[action name]>
</GTLMessages>
```

Attributes: See Table 3.

Table 3: Attributes for Task Action Messages

Attribute	Description
BrokerId	The task's broker ID. This is a uniquely generated ID for the task within an iWD instance.
CaptureId	The task's ID in the originating system.
Actor (optional)	The user or system that triggered the message. This is a free-form text field that is used for auditing purposes and will be set to name of the capture point, if none is provided.
Reason (optional)	The reason that the message was submitted. This is a free-form text field that is used for auditing purposes.
ActionDateTime (optional)	The date/time when the action was triggered. This will be set to the current date/time that the message is processed, if none is provided.

Response messages: Action-specific response messages or error messages (see "Error" on page 32).
Error codes: See Table 4.

Table 4: Error Codes for Task Action Messages

Error code	Description
INVALID_FORMAT	The message is not formatted correctly.
TASK_NOT_FOUND	The task that has the requested CaptureId or BrokerId is not found. This error code can be triggered for all action messages, except for the CreateTask message (see "CreateTask" on page 30).

Task Notification

Direction: Out
Description: This section describes common attributes and responses for all outbound messages.
Format:

```
<GTLMessages>
  <[notification name]>
    <BrokerId>Integer</BrokerId>
    <CaptureId>String(64)</CaptureId>
    <CapturePointId>String(16)</CapturePointId>
    <DistributionId>String(64)</DistributionId>
    <DistributionPointId>String(16)</DistributionPointId>
```

```
      <Actor>String(255)</Actor>
      <Reason>String(255)</Reason>
      <EventDateTime>DateTime</EventDateTime>
      [attributes specific to notification]
   </[notification name]>
</GTLMessages>
```

Attributes: See Table 5.

Table 5: Attributes for Task Notification Messages

Attribute	Description
BrokerId	The task's broker ID. This is a uniquely generated ID for the task within an iWD instance.
CaptureId	The task's ID in the originating system.
CapturePointId	The service ID of the capture point by which the task was captured.
DisbtributionId	The task's ID in the distribution system.
DistributionPointId	This attribute is provided for backward compatibility.
Actor	The user or system that triggered the notification.
Reason	The reason for the notification.
EventDateTime	The date/time when the notification was triggered.

CreateTask

Direction: In

Description: Creates a new task in iWD and populates it with the provided attributes.

All attributes of this message are optional. Typically, most of the core task attributes (such as `ProcessId, Priority`, and `BusinessValue`) are calculated and assigned by iWD rules and, therefore, should be left out.

iWD generates a new, unique broker ID for each created task that is stored in the `gtl_task.id` database field and returned in the `TaskCreated` message that is sent as a response to this message, when the task is successfully created.

Format:

```
<GTLMessages>
   <CreateTask>
```

Standard action attributes, as documented in "Task Action " on page 28, except for `BrokerId`.

```
      <channel>String(32)</channel>
      <category>String(32)</category>
      <activationDateTime>DateTime</activationDateTime>
      <dueDateTime> DateTime </dueDateTime>
```

```
        <expirationDateTime> DateTime </expirationDateTime>
        <businessValue> DateTime </businessValue>
        <priority>Integer</priority>
        <processId>String(16)</processId>
        <Ext>
           <customerID>String(64)</customerID>
           <customerSegment> String(64)</customerSegment>
           <productType> String(64)</productType>
           <productSubtype> String(64)</productSubtype>
           <resultCode> String(64)</resultCode>

<sourceFirstCreatedDateTime>DateTime</sourceFirstCreatedDateTime>
           <sourceCreatedDateTime>DateTime</sourceCreatedDateTime>
           <sourceDueDateTime>DateTime</sourceDueDateTime>
           <sourceProcessType> String(64)</sourceProcessType>
           <sourceProcessSubtype> String(64)</sourceProcessSubtype>
           <sourceTenant> String(64)</sourceTenant>
        </Ext>
        <Data>
           <CustomAttribute1> String(255)</CustomAttribute1>
           ...
        </Data>
        <Hold>Boolean</Hold>
     </CreateTask>
</GTLMessages>
```

Attributes: See Table 6.

Table 6: Attributes for CreateTask Messages

Attribute	Description
Hold	Whether to hold the task initially. If `true`, the task will be created with its initial status set to `NewHeld` and will not be processed further, until a subsequent `ResumeTask` message (see “ResumeTask” on page 43).
CaptureId (optional)	If a `CaptureId` is not provided, it will be assigned to the same generated value as `BrokerId`.

See “Task Action ” on page 28 and “TaskInfo” on page 33 for descriptions of the remaining attributes.

Response message: `TaskCreated` (see “TaskCreated” on page 32).

Error code: see Table 7.

Table 7: Error codes for CreateTask Messages

Error Code	Description
TASK_ALREADY_CAPTURED	If the captures point's `checkIfCaptured` flag is enabled, iWD will check whether a task that has a given `captureId` already exists in the database. If this is the case, the task will not be captured, and an error message that has the `TASK_ALREADY_CAPTURED` code will be submitted to the `GTLOut` queue.

TaskCreated

Direction: Out

Description: The `TaskCreated` message is submitted as a response to the `CreateTask` message (see page 30) and indicates successful task creation.

Format:

```
<GTLMessages>
   <TaskCreated>
      Standard notification attributes, as documented in "Task Notification " on page 29.
   </TaskCreated>
</GTLMessages>
```

Error

Direction: Out

Description: The `Error` message is submitted as a response to iWD request messages, indicating that the requested operation has failed.

Format:

```
<GTLMessages>
   <Error>
      <Message>String</Message>
      <Code>String</Code>
      <Parameter>String</Parameter>
      <Parameter>String</Parameter>
      ...
   </Error>
</GTLMessages>
```

Attributes: See Table 8.

Table 8: Attributes for Error Messages

Attribute	Description
Message	The formatted error message.
Code	The error code (string).
Parameter	The error parameter. There can be zero, one, or multiple error parameters. The number of parameters is specific to each error code.

GetTaskInfo

Direction: In
Description: Requests task details by using the given task's capture ID or broker ID.
Format:

```
<GTLMessages>
   <GetTaskInfo>
      Standard action attributes, as documented in "Task Action " on page 28.
   </GetTaskInfo>
</GTLMessages>
```

Response message: `TaskInfo` (see "TaskInfo" on page 33).

TaskInfo

Direction: Out
Description: The `TaskInfo` message is submitted as a response to the `GetTaskInfo` message (see "GetTaskInfo") and provides detailed information about the requested task.
Format:

```
<GTLMessages>
   <TaskInfo>
      Standard action attributes, as documented in "Task Action " on page 28.
      <tenantId>String(16)</tenantId>
      <solutionId> String(16)</solutionId>
      <ContractId> String(16)</ContractId>
      <processId> String(16)</processId>
      <channel>String(32)</channel>
      <category> String(32)</category>
      <status> String(16)</status>
      <businessCalendarId> String(16)</businessCalendarId>
      <createdDateTime>DateTime</createdDateTime>
      <heldDateTime>DateTime</heldDateTime>
      <assignedDateTime>DateTime</assignedDateTime>
      <completedDateTime>DateTime</completedDateTime>
      <activationDateTime>DateTime</activationDateTime>
      <dueDateTime>DateTime</dueDateTime>
      <expirationDateTime>DateTime</expirationDateTime>
      <priority>Integer</priority>
      <reprioritizeDateTime>DateTime</reprioritizeDateTime>
      <businessValue>Integer</businessValue>
      <assignedToUser>String(64)</assignedToUser>
      <Queue>String(255)</Queue>
      <QueueType>String(16)</QueueType>
      <QueueTarget>String(255)</QueueTarget>
      <Ext>
         <customerID>String(64)</customerID>
         <customerSegment> String(64)</customerSegment>
         <productType> String(64)</productType>
         <productSubtype> String(64)</productSubtype>
         <resultCode> String(64)</resultCode>

<sourceFirstCreatedDateTime>DateTime</sourceFirstCreatedDateTime>
         <sourceCreatedDateTime>DateTime</sourceCreatedDateTime>
         <sourceDueDateTime>DateTime</sourceDueDateTime>
         <sourceProcessType> String(64)</sourceProcessType>
         <sourceProcessSubtype> String(64)</sourceProcessSubtype>
         <sourceTenant> String(64)</sourceTenant>
      </Ext>
      <Data>
         <customAttribute1> String(255)</customAttribute1>
         ...
      </Data>
   <TaskInfo>
<GTLMessages>
```

Attributes: See Table 9.

Table 9: Attributes for TaskInfo Messages

Attribute	Description
tenantId	The task’s tenant ID, as configured in iWD Manager, assigned as soon as the task is created. This attribute is submitted to the distribution system with the `GTL_tenantId` key; updates in the distribution system are ignored.
solutionId	The tasks’s solution instance ID, as configured in iWD Manager, assigned as soon as the task is created. This attribute is submitted to the distribution system with the `GTL_solutionId` key (even if it is excluded by a filter); updates in the distribution system are ignored.

Table 9: Attributes for TaskInfo Messages (Continued)

Attribute	Description
ContractId	The tasks's contract ID, as configured in iWD Manager, assigned when the task's process is identified either by iWD rules or explicitly by the task originating system. This attribute is submitted to the distribution system with the `GTL_ContractId` key; updates in the distribution system are ignored.
processId	The tasks's process ID, as configured in iWD Manager, assigned when the task's process is identified either by iWD rules or explicitly by the task-originating system. This attribute is submitted to the distribution system with the `GTL_processId` key; updates in the distribution system are ignored.
channel	The task's media channel—for example Fax, E-mail, or Webform. This attribute is submitted to the distribution system with the `GTL_channel` key; updates in the distribution system are picked up.
category	The task's category—for example `Followup`. This attribute is submitted to the distribution system with the `GTL_category` key; updates in the distribution system are picked up.
status	Task status: `New`—The task has just been created and will be processed. `NewHeld`—The task has just been created, but it will not ne processed until it is resumed. `Captured`—The task has been processed, but it is not yet prioritized. `Queued`—The task is processed and prioritized at least once. `Assigned`—The task is assigned to an agent. `Distributed`—The task is submitted to the distribution system (Genesys). `Completed`—The task is completed. `Held`—The task is held and will not be reprioritized or distributed until it is resumed. `ErrorHeld`—An error has occurred during task processing, prioritization, or distribution. Error details are stored in the `Error` custom, extended task attribute. The task can be resumed, and the iWD will attempt to process the task again. `Canceled`—The task is canceled. `Rejected`—The task has been rejected during processing. This can occur when the task is assigned to an expired contract or process.

Table 9: Attributes for TaskInfo Messages (Continued)

Attribute	Description
businessCalendarId	The ID of the business calendar that is assigned to the task, as configured in iWD Manager. **Note:** Maximum length is 16 characters
createdDateTime	The date/time when the task has been created in iWD. This attribute is submitted to the distribution system with the `GTL_dueDateTime` key; updates in the distribution system are ignored.
heldDateTime	The date/time when the task has been held (set only when task status is either `Held`, `NewHeld`, or `ErrorHeld`).
distributedDateTime	The date/time when the task was submitted to the distribution system.
assignedDateTime	The date/time when the task has been assigned.
completedDateTime	The date/time when the task has been completed.
activationDateTime	The date and time when the task becomes active; before that, it will stay queued and will not be reprioritized and distributed. If this is not set, the task becomes active instantly.
dueDateTime	The date and time by which the task should be completed, according to the service-level agreement (SLA). This attribute is submitted to the distribution system with the `GTL_dueDateTime` key; updates within the distribution system are picked up.
expirationDateTime	The date and time when the task expires and will be archived. Only tasks that have been `Canceled`, `Completed`, or `Rejected` are archived.
priority	The task priority, which is an integer value that is used to order tasks. The higher the value, the higher that the task will stand in the distribution list and the sooner that it will be distributed. This attribute is submitted to the distribution system with the `GTL_priority` key; updates in the distribution system are picked up.
reprioritizeDateTime	The date/time when the task should be reprioritized; if this is set to `null`, no more reprioritization will be done. This value is normally updated during prioritization, based on rule expressions, such as “Reprioritize in 5 minutes”.
businessValue	The business value of the task. This attribute is submitted to the distribution system with the `GTL_businessValue` key; updates in the distribution system are picked up.

Table 9: Attributes for TaskInfo Messages (Continued)

Attribute	Description
assignedToUser	The user ID to which a task is assigned, as supplied by the distribution system.
Queue	The distribution queue name.

QueueType	The type of distribution queue:
	`InteractionQueue`
	`AgentWorkbin`
	`AgentGroupWorkbin`
	`PlaceWorkbin`
	`PlaceGroupWorkbin`

QueueTarget	The queue target—for example, `Agent ID`, if the queue type is `AgentWorkbin`.
customerId	The customer's ID. This attribute is submitted to the distribution system with the `GTL_ext_customerId` key; updates in the distribution system are picked up.
customerSegment	The customer's segment or value. This attribute is submitted to the distribution system with the `GTL_ext_customerSegment` key; updates in the distribution system are picked up.
productType	The related product—for example, DSL. This attribute is submitted to the distribution system with the `GTL_ext_productType` key; updates in the distribution system are picked up.
productSubtype	The subtype of the related product—for example: PremiumDSL. This attribute is submitted to the distribution system with the `GTL_ext_productSubtype` key; updates in the distribution system are picked up.
resultCode	The task-result code/outcome; typically, it is set by an agent in a softphone or another client application. This attribute is submitted to the distribution system with the `GTL_ext_resultCode` key; updates in the distribution system are picked up.
sourceFirstCreatedDateTime	The earliest timestamp of the task in the enterprise; it is applicable if there is another system, such as a fax server, that is used before the task-originating system. This attribute is submitted to the distribution system with the `GTL_ext_sourceFirstCreatedDateTimeTime` key; updates in the distribution system are ignored.

Table 9: Attributes for TaskInfo Messages (Continued)

Attribute	Description
sourceCreatedDateTime	The task-creation timestamp in the task-originating system. This attribute is submitted to the distribution system with the `GTL_ext_sourceCreatedDateTime` key; updates in the distribution system are ignored.
sourceDueDateTime	The task-due timestamp in the task-originating system. This attribute is submitted to the distribution system with the `GTL_ext_sourceDueDateTime` key; updates in the distribution system are ignored.
sourceProcessType	A related process in the task-originating system—for example, Order. This attribute is submitted to the distribution system with the `GTL_ext_sourceProcessType` key; updates in the distribution system are ignored.
sourceProcessTypeSubtype	The subtype of the related process in the task-originating system. This attribute is submitted to the distribution system with the `GTL_ext_sourceProcessSubtype` key; updates in the distribution system are ignored.
sourceTenant	The tenant ID or name in the task originating system. This attribute is submitted to the distribution system with the `GTL_ext_sourceTenant` key; updates in the distribution system are ignored.
data	Custom task attributes. These attributes can be used to associate additional task originating system-specific data to the task that can be used in iWD rules, routing, and historical reporting.

UpdateTask

Direction: In

Description: Updates the attributes of the task that has the given task's capture ID or broker ID. This results in the interaction properties of the task being updated through the Genesys Interaction Server.

All attributes except for `CaptureId` and `BrokerId` are optional. If the attribute is not provided, it will not be updated.

Format:

```
<GTLMessages>
   <UpdateTask>
```

Standard notification attributes, as documented in "Task Action " on page 28.

```
      <category>String(32)</category>
      <activationDateTime>DateTime</activationDateTime>
      <dueDateTime> DateTime </dueDateTime>
      <expirationDateTime> DateTime </expirationDateTime>
      <businessValue>Integer</businessValue>
      <priority>Integer</priority>
      <ext>
         <customerID>String(64)</customerID>
         <customerSegment> String(64)</customerSegment>
         <productType> String(64)</productType>
         <productSubtype> String(64)</productSubtype>
         <resultCode> String(64)</resultCode>

<sourceFirstCreatedDateTime>DateTime</sourceFirstCreatedDateTime>
         <sourceCreatedDateTime>DateTime</sourceCreatedDateTime>
         <sourceDueDateTime>DateTime</sourceDueDateTime>
         <sourceProcessType> String(64)</sourceProcessType>
         <sourceProcessSubtype> String(64)</sourceProcessSubtype>
         <sourceTenant> String(64)</sourceTenant>
      </ext>
      <data>
         <customAttribute1> String(255)</customAttribute1>
         ...
      </data>
   </UpdateTask>
</GTLMessages>
```

Attributes: See “Task Action ” on page 28 and “TaskInfo” on page 33 for a description of the attributes.

Response message: “TaskUpdated”.

TaskUpdated

Direction: Out

Description: The `TaskUpdated` message is submitted as a response to the `UpdateTask` message (see “UpdateTask” on page 38), as well as when the task is updated either by iWD Manager or within the distribution system.

Format:

```
<GTLMessages>
   <TaskUpdated>
```

Standard notification attributes, as documented in“Task Notification ” on page 29.

```
      <tenantId>String(16)</tenantId>
      <solutionId> String(16)</solutionId>
      <ContractId> String(16)</ContractId>
      <processId> String(16)</processId>
      <channel>String(32)</channel>
      <category> String(32)</category>
      <status> String(16)</status>
      <businessCalendarId> String(16)</businessCalendarId>
      <createdDateTime>DateTime</createdDateTime>
      <heldDateTime>DateTime</heldDateTime>
      <assignedDateTime>DateTime</assignedDateTime>
      <completedDateTime>DateTime</completedDateTime>
      <activationDateTime>DateTime</activationDateTime>
      <dueDateTime>DateTime</dueDateTime>
      <expirationDateTime>DateTime</expirationDateTime>
      <priority>Integer</priority>
      <reprioritizeDateTime>DateTime</reprioritizeDateTime>
      <businessValue>Integer</businessValue>
      <assignedToUser>String(64)</assignedToUser>
      <ext>
         <customerID>String(64)</customerID>
         <customerSegment> String(64)</customerSegment>
         <productType> String(64)</productType>
         <productSubtype> String(64)</productSubtype>
         <resultCode> String(64)</resultCode>

<sourceFirstCreatedDateTime>DateTime</sourceFirstCreatedDateTime>
         <sourceCreatedDateTime>DateTime</sourceCreatedDateTime>
         <sourceDueDateTime>DateTime</sourceDueDateTime>
         <sourceProcessType> String(64)</sourceProcessType>
         <sourceProcessSubtype> String(64)</sourceProcessSubtype>
         <sourceTenant> String(64)</sourceTenant>
      </ext>
      <data>
         <customAttribute1> String(255)</customAttribute1>
         ...
      </data>
      <actor>String(255)</actor>
   <TaskUpdated>
<GTLMessages>
```

Attributes: See “Task Action ” on page 28 and “TaskInfo” on page 33 for a description of the attributes.

TaskDistributed

Direction: Out

Description: The `TaskDistributed` message is submitted when the task is distributed by the distribution system (that is, Genesys).

Format:

```
<GTLMessages>
   <TaskDistributed>
```

Standard notification attributes, as documented in “Task Notification ” on page 29.

```
   </TaskDistributed>
</GTLMessages>
```

Attributes: See “Task Notification ” on page 29 and “TaskInfo” on page 33 for descriptions of the attributes.

TaskDistributedQueue

Direction: Out

Description: The `TaskDistributedQueue` message is submitted when the task is moved by the distribution system (that is, Genesys) into a routing queue or workbin.

Format:

```
<GTLMessages>
   <TaskDistributedQueue>
      Standard notification attributes, as documented in “Task Notification ”
on page 29.
      <Queue>String(255)</Queue>
      <QueueType>String(16)</QueueType>
      <QueueTarget>String(255)</QueueTarget>
   </TaskDistributedQueue>
</GTLMessages>
```

Attributes: See “Task Notification ” on page 29 and “TaskInfo” on page 33 for a description of the attributes.

TaskAssigned

Direction: Out

Description: The `TaskAssigned` message is submitted when the task is assigned to an agent.

Format:

```
<GTLMessages>
   <TaskAssigned>
   Standard notification attributes, as documented in “Task Notification ” on
page 29.
      <AssignedToUser>String(64)</AssignedToUser>
   </TaskAssigned>
</GTLMessages>
```

Attributes: See “Task Notification ” on page 29 and “TaskInfo” on page 33 for a description of the attributes.

CompleteTask

Direction: In

Description: Completes the task that has a given capture ID or broker ID.

Format:

```
<GTLMessages>
   <CompleteTask>
      Standard action attributes, as documented in“Task Action ” on page 28.
   </CompleteTask>
</GTLMessages>
```

Attributes: See “Task Action ” on page 28 for a description of the attributes.

Response message: TaskCompleted. See “TaskCompleted” on page 42.

Error codes: See Table 10.

Table 10: Error Codes for CompleteTask Messages

Error code	Description
CANNOT_COMPLETE_TASK	Cannot complete the task, because it is already completed, canceled, or rejected.

TaskCompleted

Direction: Out

Description: The `TaskCompleted` message is submitted as a response to the `CompleteTask` message (see “CompleteTask” on page 41), as well as when the task is placed into the predefined `CompletedQueue` interaction queue in Interaction Server.

Format:

```
<GTLMessages>
   <TaskCompleted>
      Standard notification attributes, as documented in “Task Notification ”
on page 29
   </TaskCompleted>
</GTLMessages>
```

Attributes: See “Task Notification ” on page 29 for a description of the attributes.

HoldTask

Direction: In

Description: Holds the task that has given task’s capture ID or broker ID. As soon as it is held, the task will not be reprioritized or, potentially, assigned until it is resumed (see “ResumeTask” on page 43).

Only tasks that are not held, assigned, completed, canceled, or rejected can be held.

Format:

```
<GTLMessages>
   <HoldTask>
      Standard action attributes, as documented in “Task Action ” on page 28.
   </HoldTask>
</GTLMessages>
```

Attributes: See “Task Action ” on page 28 for a description of the attributes.

Response message: `TaskHeld` (see “TaskHeld” on page 43).

Error codes: See Table 11.

Table 11: Error Codes for HoldTask Messages

Error code	Description
CANNOT_HOLD_TASK	Cannot hold the task because it is assigned, completed, canceled, rejected, or already held.

TaskHeld

Direction: Out

Description: The `TaskHeld` message is submitted as a response to the `HoldTask` message (see “HoldTask” on page 42), as well as when the task is held from the iWD Manager.

Format:

```
<GTLMessages>
   <TaskHeld>
      Standard notification attributes, as documented in “Task Notification ” on page 29.
   </TaskHeld>
</GTLMessages>
```

Attributes: See “Task Notification ” on page 29 for a description of the attributes.

TaskErrorHeld

Direction: Out

Description: The `TaskErrorHeld` message is submitted when the task is held because of a configuration error (such as incomplete rules).

Format:

```
<GTLMessages>
   <TaskErrorHeld>
      Standard notification attributes, as documented in “Task Notification ” on page 29.
      <Error>String(255)</Error>
   </TaskErrorHeld>
</GTLMessages>
```

Attributes: See “Task Notification ” on page 29 for a description of the attributes.

ResumeTask

Direction: In

Description: Resumes the held task that has the given task’s capture ID or broker ID.

As soon as it is resumed, the task will be processed and assigned normally, according to the iWD rules.

Only tasks that are held can be resumed.

Format:

```
<GTLMessages>
   <ResumeTask>
      Standard action attributes, as documented in “Task Action ” on page 28.
   </ResumeTask>
</GTLMessages>
```

Attributes: See “Task Action ” on page 28 for a description of the attributes.

Response message: `TaskResumed`. See “TaskResumed” on page 44.

Error codes: See Table 12.

Table 12: Error Codes for ResumeTask Messages

Error code	Description
CANNOT_RESUME_TASK	Cannot resume the task, because it is not held.

TaskResumed

Direction: Out

Description: The `TaskResumed` message is submitted as a response to the `ResumeTask` message (see “ResumeTask” on page 43), as well as when a task is held from the iWD Manager.

Format:

```
<GTLMessages>
   <TaskResumed>
      Standard notification attributes, as documented in “Task Notification ”
on page 29.
   </TaskResumed>
</GTLMessages>
```

Attributes: See “Task Notification ” on page 29 for a description of the attributes.

RestartTask

Direction: In

Description: Restarts the task that has the given task’s capture ID or broker ID.

As soon as it is restarted, the task will be reclassified and reprioritized.

Only tasks that are not assigned, completed, canceled, or rejected can be restarted.

Format:

```
<GTLMessages>
   <RestartTask>
      Standard action attributes, as documented in “Task Action ” on page 28.
   </RestartTask>
</GTLMessages>
```

Attributes: See “Task Action ” on page 28 for a description of the attributes.

Response message: `TaskRestarted` (see “TaskRestarted” on page 45).

Error codes: See Table 13.

Table 13: Error codes for RestartTask Messages

Error Code	Description
CANNOT_RESTART_TASK	Cannot restart task because it is assigned, canceled, completed, or rejected.

TaskRestarted

Direction: Out

Description: The `TaskRestarted` message is submitted as a response to the `RestartTask` message (see “RestartTask”), as well as when the task is either restarted from the iWD Manager or is moved to the pre-defined `New` interaction queue within the distribution system.

Format:

```
<GTLMessages>
   <TaskRestarted>
      Standard notification attributes, as documented in “Task Notification ”
on page 29.
   </TaskRestarted>
</GTLMessages>
```

Attributes: See “Task Notification ” on page 29 for a description of the attributes.

CancelTask

Direction: In

Description: Cancels the task that has the given task’s capture ID.

As soon as it is canceled, task processing will be completely halted.

Only tasks that are not assigned, completed, or rejected can be canceled.

Format:

```
<GTLMessages>
   <CancelTask>
      Standard action attributes, as documented in “Task Action ” on page 28.
   </CancelTask>
</GTLMessages>
```

Attributes: See “Task Action ” on page 28 for a description of the attributes.

Response message: `TaskCanceled` (see “TaskCanceled”, below).

Error codes: See Table 14.

Table 14: Error Codes for CancelTask Messages

Error code	Description
CANNOT_CANCEL_TASK	Cannot cancel task, because it is completed, canceled or rejected.

TaskCanceled

Direction: Out

Description: The `TaskCanceled` message is submitted as a response to the `CancelTask` message (see "CancelTask"), as well as when the task is canceled from iWD Manager.

Format:

```
<GTLMessages>
   <TaskCanceled>
      Standard notification attributes, as documented in "Task Notification "
on page 29.
   </TaskCanceled>
</GTLMessages>
```

Attributes: See "Task Notification " on page 29 for a description of the attributes.

TaskRejected

Direction: Out

Description: The `TaskRejected` message is submitted when the task is rejected by the iWD Classification Service. The task can be rejected when a process or department to which the task is assigned is currently inactive (that is, either expired or not yet active).

Format:

```
<GTLMessages>
   <TaskRejected>
      Standard notification attributes, as documented in "Task Notification "
on page 29.
   </TaskRejected>
</GTLMessages>
```

Attributes: See "Task Notification " on page 29 for a description of the attributes.

Ping

Direction: In

Description: A simple `Ping` message that can be used to check the health of the JMS Capture Point. The message includes an ID, which will be present in the corresponding `Pong` message.

Format:

```
<GTLMessages>
   <Ping>ID</Ping>
</GTLMessages>
```

Response message: `Pong` (see “Pong”).

Pong

Direction: Out

Description: Submitted as a response to the `Ping` message (see “Ping” on page 46), indicating that the JMS Capture Point service is active.

The `Pong` message contains the ID that was sent in the `Ping` message.

Format:

```
<GTLMessages>
   <Pong>ID</Pong>
</GTLMessages>
```

Index