



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

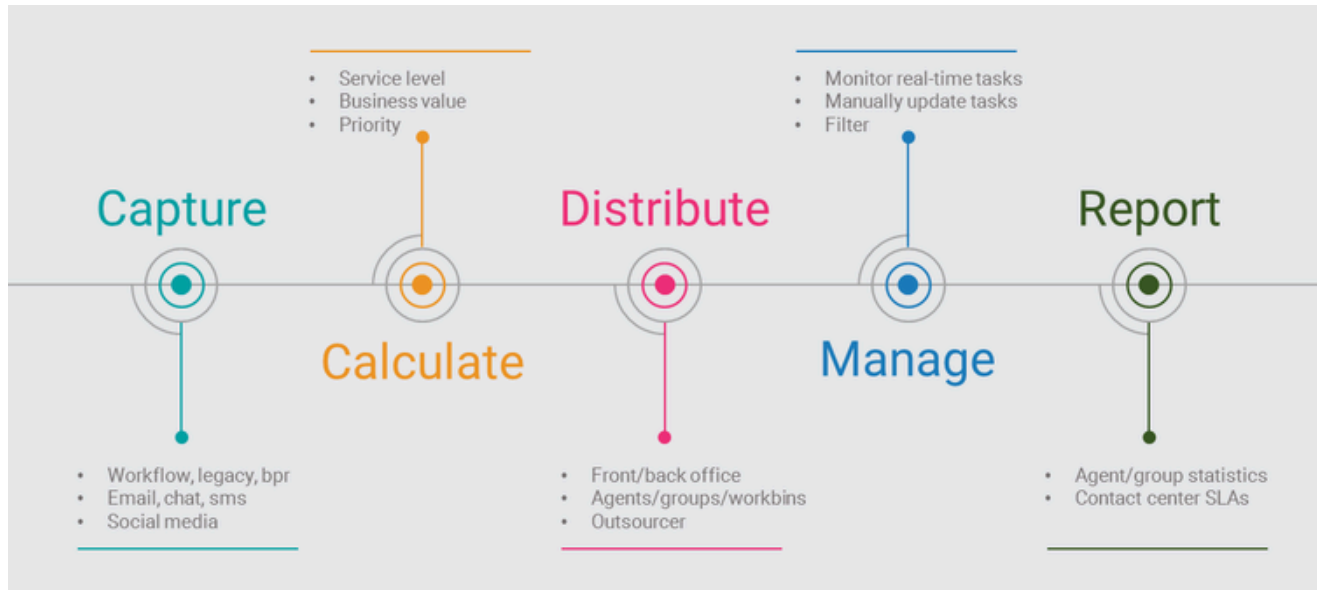
Working with iWD Capture Points

intelligent Workload Distribution 8.5.1

Table of Contents

Working with iWD Capture Points	3
Capture Point Messages	7
iWD Task Data Model	10
Web Service Capture Point	13
Database Capture Point	17
JMS Capture Point	22
XML File Capture Point	25
Common Features	29
Selecting an Appropriate Capture Point	31

Working with iWD Capture Points



This document discusses how iWD **captures** work from an enterprise's source systems. It introduces the concept of the *capture adapter* (or *capture point*), which iWD uses to generalize integration with source systems in a coherent, repeatable way in order to reduce IT costs and project timelines. It describes the iWD message set and the iWD task data model. It then presents information about the four types of iWD capture points available, with enough detail to help IT architects understand how to assess which capture point is the best fit for each source system and how to accomplish integration.

This video introduces integrated capture points from an eServices perspective:

[Link to video](#)

What is an iWD Capture Point?

An iWD capture point, also referred to as a capture adapter, provides a standardized way for the many source systems in an enterprise to communicate with the iWD solution. The communication will take different forms depending on the specific capture point used, but the operations performed by each capture point will be similar. These operations include:

- Creating a new task in the iWD system, based on some existing piece of work owned by an enterprise source system.
- Canceling an existing task.
- Putting a task on hold.
- Resuming and held task.

- Completing a task.
- Updating specific task data.

All iWD capture points are integrated into the Genesys Interaction Server software component. The Interaction Server is the main component in the iWD system and is responsible for managing the entire lifecycle of tasks, from the time they are captured from the source systems to the time they are eventually removed from the Genesys system. The capture points are exposed by Interaction Server as methods by which external (source) systems can communicate with iWD.

Types of Capture Point

Four types of iWD capture points are described in this document:

- Web Service
- Database
- Java Message Service (JMS)
- XML File

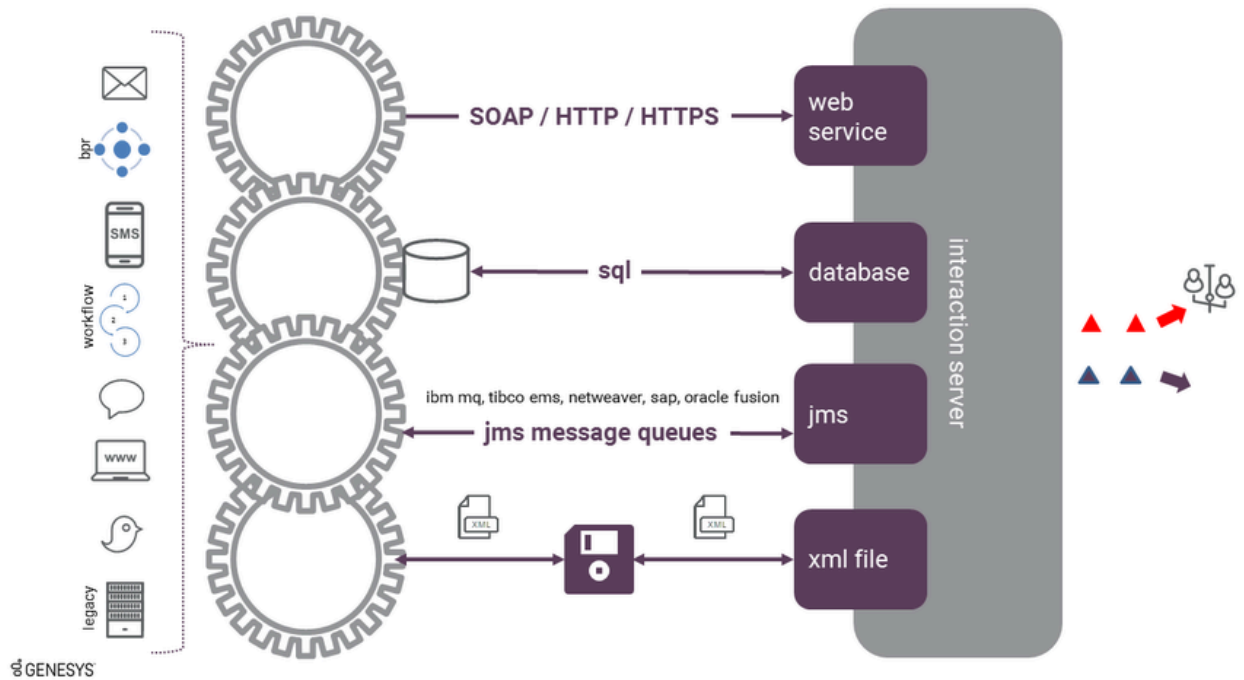
Using one of these four capture points, you can set up communication between iWD and virtually any source system in the enterprise. The capture point to use will depend on the integration capabilities of the source system and other factors such as whether it is important to have bi-directional messaging between iWD and the source system. See [Selecting an Appropriate Capture Point](#) for more detail about the criteria to consider when selecting which capture point to use.

Important

In release 9.0, the iWD Web component has special in-built capture point functionality that enables it to handle automatic upload to iWD of spreadsheet files that contain tasks generated by external source systems. To read more about this functionality see the [Deployment Guide](#).

Capture Point Architecture

This diagram shows the four different iWD capture points and how they relate to the source systems. The specific features of each capture adapter, the message format, and other details are provided later in this document.



Messaging

For the three iWD capture points that support a specific message format—that is, all the capture points other than the Database Capture Point—the latest releases support backwards-compatibility with the message format of the legacy capture points.

Important

Legacy capture points are not natively supported from release 8.5 onwards.

Specifically for the JMS and XML File Capture Points, which support the same XML message schema, backwards-compatibility is provided with the legacy iWD capture point message format through the use of XML message transformation. An XML message in iWD-compatibility format can be submitted to the capture point, and out-of-the-box scripts will be executed to transform the XML into Interaction Server’s native message format. In the case of the Web Service Capture Point, the integration with a source system may be done using one of two supplied Web Service Description Languages (WSDLs)—either the iWD-compatibility WSDL, or the native Interaction Server WSDL. The iWD-compatibility WSDL is the way that the Web Service Capture Point supports backwards compatibility with the legacy capture point message format.

Where this document describes the capture point messages, the iWD-compatibility messages are used. The actual differences between the iWD-compatibility and native Interaction Server messages are not substantial, but the iWD-compatibility messages are a bit more descriptive since they map

directly to the iWD task lifecycle. For instance, the iWD-compatibility XML messages and WSDL have requests such as `CompleteTask` and `CancelTask`, whereas the native Interaction Server XML message set and WSDL use an update request to instruct Interaction Server to move a task from to a new queue that represents the task status—for example to `iWD_Completed` or `iWD_Canceled`.

To keep things simple, this document uses the generic term *task attributes* to describe the data elements of an iWD task that can be set, updated, or queried, through a capture point. In fact, the iWD task attributes are a subset of all the Interaction Server interaction properties that can be accessed and manipulated. These nuances are well covered in the Genesys product documentation and can be studied prior to embarking on the design of any source system-iWD integration project.

Further Reading

- [eServices Integrated Capture Points Guide](#) has detailed technical reference and configuration information about setting up your capture points.
- [Working with Integrated Capture Points](#) in the iWD Deployment Guide. This includes information about compatibilities with legacy capture points.

Capture Point Messages

There are eight main messages supported by the iWD capture points. For all of these messages other than Create Task, it is possible to use one of two different identifiers for the task:

- The Interaction Id (also sometimes referred to as the Task Id)—An identifier generated by the Genesys Interaction Server and therefore guaranteed to be unique within the iWD system.
- The Capture Id—Normally the Capture Id is mapped directly to an identifier of the corresponding work item on the source system, such as a Siebel SR number. The Capture Id is stored by Interaction Server as an interaction property with the key ExternalId.

Submit Task

This message is used to create a new task in iWD. When a new task is created, many task attributes can be set, based on data coming from the source system. The main purpose of passing such data along with the Create Task message is for iWD to use this data in its business rules, in order to classify, prioritize, and distribute the task to the appropriately skilled employee. For example, the data record for a work item in the source system may contain information about:

- A product or service that was ordered.
- An ID of the work item
- The customer's status with the company
- The original due date for the work item.

All of this information may be valuable to pass into iWD to be used as criteria to determine how to appropriately prioritize the task in iWD, how to select the right skilled employee, what due date to give to the task, and so on.

Update Task

This message is used to update or change task attributes in iWD, which means that the value of specified task attributes will be modified. Normally this message is sent to iWD from the source system based on some event that occurred on the source system itself, requiring that iWD be informed of the most up-to-date values of a data attribute, or attributes, that changed. For example, a customer record may be updated on the source system, which in turn initiates an update to a particular work item. Perhaps the due date of a work item was updated because a customer called into the contact center, and it is important to propagate this date to iWD to update the task that has not yet been processed. Such an update may have important implications for how the priority of the task is calculated within iWD.

Restart Task

This message is used to return a task to the starting point of the iWD business process. All tasks that are created in iWD flow through a simple business process that manages the lifecycle of the task in iWD. This business process defines the points at which business rules are invoked on the task, for example to classify it and prioritize it. The business process also defines the criteria used for the Genesys routing engine to distribute the task to the right employee. By restarting the task, the source system is instructing iWD to return the task back to the beginning of this business process, which will allow the task to be re-classified and then go through the remaining business rules to be prioritized. A use case for this would be if there was an error in the processing of the task, either in iWD or on the source system. This may occur for example if some critical piece of data is missing in the work item's data record that caused it to be mis-classified or not classified at all (and thus, placed into an error state). Once this missing piece of data is added, the processing of the task within iWD may be re-initiated by using this Restart Task message.

Hold Task

This message is used to temporarily suspend the processing of a task in iWD. It may be used in cases where something occurs on the source system that necessitates that the task not be distributed to an employee. For example, some key information is missing from the work item's data record and this information must be provided on the source system before resuming the processing of the associated task in iWD.

Resume Task

This message is used to resume processing of a held task.

Stop Info

This message is used to Stop or delete a task. Stopping a task means that the task is removed from the running interactions task list and is deleted from the Interaction Server database. Only existing, running, or held interactions can be stopped.

Get Task Info

This message is used to query iWD to return all the information that is known about a task. This includes all the attributes of the task as well as its status, and the ID of the employee who is working on the task, if it has been assigned. The primary reason to use this message is when using a capture point such as the Web Service Capture Point, which is not bi-directional in nature and therefore will not generate unsolicited notifications to the source system when a task is assigned to an employee or a business rule updates the value of some task attribute. The notification returned by this request is known as the Task Info message.

Replies & Notifications Sent by the Capture Points

Depending on the capture point being used, the messages will return different replies. For example, if the Submit Task message is used to create a new task in the iWD system, upon success the reply returned to the source system from iWD will be the ID of the task. If the request fails, an error number and message will be returned. In addition, three of the capture points—the JMS, XML File, and Database Capture Points—will provide unsolicited notifications back to the source system, when certain events occur in the iWD system. For example, if an employee is assigned to a task in iWD, a notification will be generated back to the source system informing the system of the ID of this employee. These notifications can be filtered based on configuration parameters, so that only the notifications of interest will be sent by the capture point.

iWD Task Data Model

iWD supports a well-defined data model that allows task data within iWD to be mapped to the data model of the source system(s) with which iWD integrates. The purpose of the iWD task data model is to provide a way to unify disparate data models that may exist across multiple source systems and provide a common way to understand the data as it will appear in iWD business rules and iWD management views and reports. To accomplish this, iWD's data model separates task data into three types of attributes:

- Core attributes
- Extended attributes
- Custom attributes

Core attributes

Core attributes are used to describe the fundamental elements of a task in iWD. Generally, the value of each attribute is either set automatically by iWD through business rules, or provided by the source system through the capture point interface. (It is possible, however, to set and/or update attribute values in other ways such as directly from an employee desktop application, or from a Genesys routing strategy.) Table 1 lists the iWD core task attributes. The Direction column specifies In/Out when the attribute can be both set as part of a Submit Task or Update Task message or Out when the attribute cannot be set or updated directly in a capture point message, but is returned by the capture point when the source system issues a Get Task Info request.

Core Task Attribute	Direction
Activation Date/Time	In/Out
Assigned Date/Time	Out
Assigned to User	Out
Business Calendar ID	Out
Business Value	In/Out
Capture ID	In/Out
Category	In/Out
Channel	In/Out
Completed Date/Time	Out
Created Date/Time	In/Out
Department ID	Out
Due Date/Time	In/Out
Expiration Date/Time	In/Out
Held Date/Time	Out
Interaction ID	Out

Core Task Attribute	Direction
Media Type Channel	In/Out
Priority	In/Out
Process ID	In/Out
Queue	Out
Queue Target	Out
Queue Type	Out
Reprioritize Date/Time	Out
Solution ID	Out
Status	Out
Tenant ID	Out

Extended attributes

Extended attributes provide additional context about a task and can aid in customizing current-day and historical reporting. For example, use of several capture dates allows an organization to measure performance against the date and time at which work item was received by the source system or was submitted by the customer via a web form. The table below lists the iWD extended task attributes.

Extended attribute	Direction
Customer ID	In/Out
Customer Segment	In/Out
Product Type	In/Out
Product Subtype	In/Out
Requested Agent	In/Out
Requested Agent GroupExample	In/Out
Requested Place Group	In/Out
Requested Skill	In/Out
Result Code	In/Out
Source Created Date/Time	In/Out
Source Due Date/Time	In/Out
Source First Created Date/Time	In/Out
Source Process Type	In/Out
Source Process Subtype	In/Out
Source Tenant	In/Out

Custom attributes

Custom attributes enable you to customize additional task details. Custom attributes are key-value pairs that are generally provided by the source system. For example, a web form collecting data for an enterprise source system can contain several fields that might not be mapped to a core or extended attribute. Instead, they can be mapped to custom attributes. Custom attributes can be used in reporting, even as dimensions in the iWD Data Mart schema. As part of the planning process for an iWD deployment, the various data elements from the source systems must be mapped to the corresponding task attributes in iWD. In general, core and extended attributes are preferred over custom attributes, since the core and extended attributes are well supported for business rules, task management, and reporting, without any additional configuration.

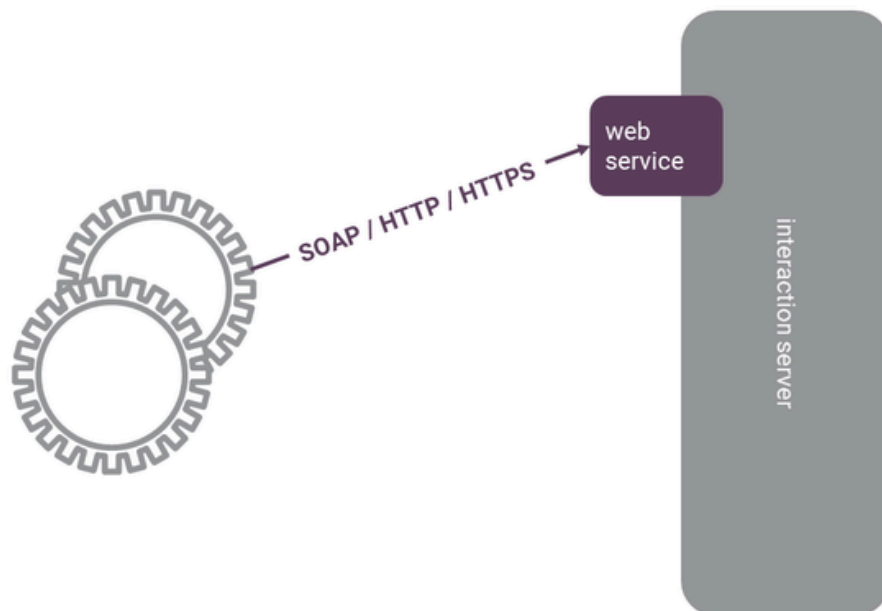
Further Reading

- [Working with Task Attributes and Properties](#) in the iWD Deployment Guide contains detailed reference, mapping and configuration information for working with iWD task attributes and properties.

Web Service Capture Point

The Web Service Capture Point allows a source system to communicate with the iWD system via a SOAP-based Web service. The definition of the Web service is exposed in Web Services Description Language (WSDL) format. There are two WSDLs that can be exposed, based on the setting of a configuration parameter (**iwd-compatibility-mode**). The iWD-compatibility WSDL exposes an interface that provides backwards compatibility with the iWD legacy Web Service Capture Point, and the native WSDL exposes an interface that uses slightly different messages and message structures, but supports very similar functionality.

The Web Service Capture Point allows a source system to create a new task in iWD, as well as cancel, complete, hold, resume, update, or restart a task. In addition, iWD task data may also be updated through the Web Service Capture Point. A heartbeat monitoring—or ping—message is also supported.



GENESYS

The Web Service Capture Point is not bi-directional. That is, if updates occur to a task on iWD, such as a task being assigned to an employee, or otherwise changing status (being completed, canceled, or rejected), or when one or more of its attributes is modified, these notifications will not be propagated to the source system. It is the responsibility of the source system to query iWD through the Get Task Info message of the Web Service Capture Point in order to obtain the latest information about the task.

Some of the additional features of the Web Service Capture Point include:

- Support for proxy servers to address security requirements

- Support for secure HTTP based on customer-generated keys

The Web Service Capture Point has been tested against several development environments, using the WSDL that is exposed to generate client proxies in both Java and .NET. The development environments tested with include Microsoft Visual Studio 2010, JAX-WS, Apache CXF, and Apache Axis2/Java. The product documentation includes step-by-step instructions describing how these environments can be used to generate Web service clients.

Example Web Service Capture Point Request

Below is an example of a simple request using the iWD Web Service Capture Point, which instructs iWD to create a new task. This request includes examples of iWD core, extended, and custom attributes. The core attributes are those such as captureId, which can be seen near the top of the request. The custom attributes are the key-value pairs that are enclosed within the < UserData ></ UserData > elements.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:ixn="http://www.genesyslab.com/interaction">
  <SOAP-ENV:Body>
    <ixn:Submit xmlns="http://www.genesyslab.com/interaction">
      <TenantId>101</TenantId>
      <Queue>Queue1</Queue>
      <ExternalId>Test00001</ExternalId>
      <UserData>
        <kvitem><key>StringKey</key><value><ValueString>StringValue</ValueString></value></kvitem>
        <kvitem><key>IntKey</key><value><ValueInt>812</ValueInt></value></kvitem>
        <kvitem><key>List1Key</key><value><ValueList>
          <kvitem><key>StringKeyL1</key><value><ValueString>StringValueL1</ValueString></value></kvitem>
          <kvitem><key>IntKeyL1</key><value><ValueInt>1812</ValueInt></value></kvitem>
          <kvitem><key>List2Key</key><value><ValueList>
            <kvitem><key>StringKeyL2</key><value><ValueString>StringValueL2</ValueString></value></kvitem>
            <kvitem><key>IntKeyL1</key><value><ValueInt>11812</ValueInt></value></kvitem>
          </ValueList></value></kvitem>
        </ValueList></value></kvitem>
      </UserData>
    </ixn:Submit>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Integration

To use the Web Service Capture Point, the source system must be able to invoke the iWD web service in order to take some action as described earlier in this document. Actually generating the web service client proxy to be invoked from within the source system's workflow or EAI component should not be difficult, and detailed examples are provided in the [Web Service Capture Point](#) section of the eServices Integrated Capture Points Guide. Once these client proxies are generated, the iWD web service can be invoked from the appropriate places in the source system's workflow component in order to create tasks, cancel tasks, complete tasks, and so on. In the case of completing or canceling a task, or even getting task info, the message is very simple because the web service call just needs to include an identifier of the task of interest. The Create Task message is the most complex because it requires that the SOAP message be formatted as described earlier, and contain certain attributes that are necessary to create a new task in iWD.

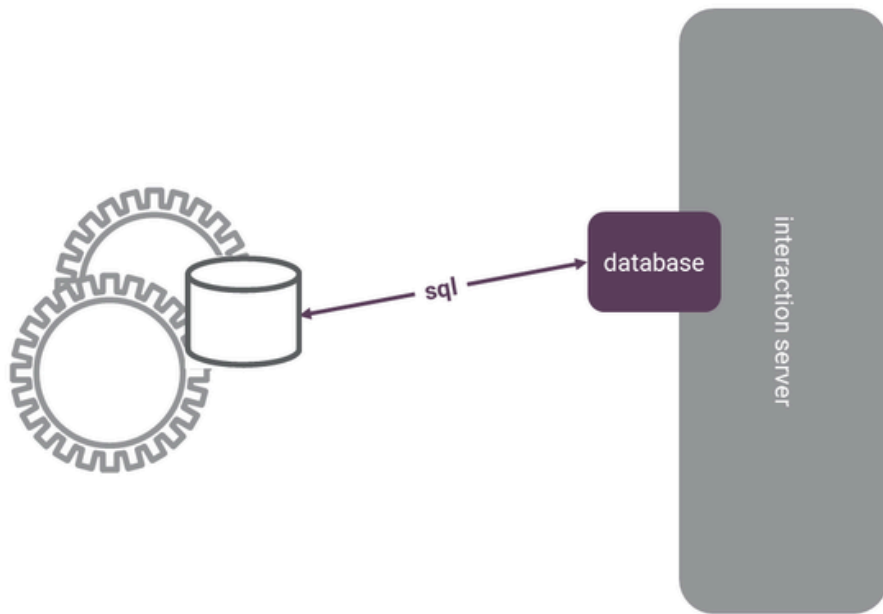
There are tools such as SOAP UI that allow you to point to the iWD Web Service Capture Point WSDL and automatically generate sample requests for each message. These samples can then be used for simulating and testing the various use cases of the source system interacting with iWD.

Further Reading

There is more detailed technical reference information for this capture point in the [Web Service Capture Point](#) section of the eServices Integrated Capture Point Reference Guide.

Database Capture Point

The Database Capture Point provides the ability to create new tasks by directly accessing and querying a source system's database. It also provides the ability to query the source system's database to look for records that have been updated, to then propagate the updates to the corresponding tasks in iWD. Moreover, it is possible to configure database queries that will update specific tables on the source system database when updates to a task's status or other attributes occur in iWD.



GENESYS

The Database Capture Point connects directly to each source system database via ODBC. The configuration of this capture point includes the ODBC connection string or data source name configuration (after setting up a DSN) and the SQL queries that the capture point should execute based on each iWD event. The syntax used for the SQL queries should observe the semantics of the specific database server being used (Microsoft SQL Server, Oracle, and IBM DB2 are supported).

Supported Queries

The Database Capture Point supports the following queries. The tables below show the name of each query as it appears in the capture point configuration options, an explanation of what iWD event corresponds to this query, and an example of the query syntax. The tables are separated into the Inbound Queries, Notification Queries, and Source Update Queries.

Note that when using parameters in a query (for example, "externalid=<external id of the task>"), you write a question mark followed by the name of the parameter as known to the Genesys Interaction Server (which is referred to as an interaction property), in single quotes. The parameter names and interaction property names are case-sensitive.

Inbound Queries

Inbound Query	Corresponding iWD Event	Example
captureQuerySql	This query returns a result set in which each row will be captured as a task in iWD. The result set may contain columns corresponding to task attributes. There must be a field in the table or view that indicates if the task has already been captured; in this example that field is status.	select caseId "ExternalId", createdTimestamp "ReceivedAt", casePriority "Priority", status "Status" from cases where status='new'
capturedUpdateSql	This query updates the corresponding database record to reflect that certain data has been successfully captured as a task in iWD. Besides the values available from the corresponding capture query, the InteractionId (a unique value assigned by iWD to each task) is available to this query. This update will be executed for every task captured in the captureQuerySql set	update cases set caseTrackingId=?'InteractionId', status='submitted' where caseId=?'ExternalId'
errorUpdateSql	This query updates the corresponding database record to reflect that the associated task has not been captured in iWD. Besides the values available from the corresponding capture query, the additional values ErrorCode (integer) and ErrorDescription (string up to 256 characters) are available to this query.	update cases set status='error', errorCode=?'ErrorCode', errorDescr=?'ErrorDescription' where caseId=?'ExternalId'

Notification Queries

For all of the Notification Queries described below, it is possible to use the value of any iWD task attribute to update the corresponding column in the source system database.

Notification Query	Corresponding iWD Event	Example
assignedUpdateSql	This query updates the database to reflect that the associated task has been put in an Assigned status in iWD (i.e. assigned to an employee).	update cases set status = 'assigned' where caseId=?'ExternalId'
completedUpdateSql	This query updates the database to reflect that the associated task has been put in a Completed status in iWD.	update cases set status = 'completed' where caseId=?'ExternalId'
canceledUpdateSql	This query updates the database to reflect that the associated task has been put in a Canceled status in iWD.	update cases set status = 'canceled' where caseId=?'ExternalId'
heldUpdateSql	This query updates the database to reflect that the associated task has been put in a Held status in iWD.	update cases set status = 'held' where caseId=?'ExternalId'

Notification Query	Corresponding iWD Event	Example
queuedUpdateSql	This query updates the database to reflect that the associated task has been placed into any queue in the iWD business process, other than the standard queues that are used when tasks are canceled, completed, rejected, or put into an error status.	update cases set status = 'queued', substatus=?'Queue' where caseId=?'ExternalId'
errorHeldUpdateSql	This query updates the database to reflect that the associated task has been put in an ErrorHeld status in iWD.	update cases set status = 'errorheld' where caseId=?'ExternalId'
rejectedUpdateSql	This query updates the database to reflect that the associated task has put into a Rejected status in iWD.	update cases set status = 'rejected' where caseId=?'ExternalId'
restartedUpdateSql	This query updates the database to reflect that the associated task has been restarted in the iWD business process (to be reclassified and reprioritized).	update cases set status = 'restarted' where caseId=?'ExternalId'
stoppedUpdateSql	This query updates the database to reflect that the associated task has been stopped in the Genesys Interaction Server. This means that the task gets deleted completely from the iWD system.	update cases set status = 'stopped' where caseId=?'ExternalId'
routeRequestedUpdateSql	This query updates the database to reflect that the associated task has been sent to the Genesys Universal Routing Server to request that it be routed to an employee.	update cases set status = 'routing' where caseId=?'ExternalId'
updatedUpdateSql	This query updates the database to reflect that the associated task has been updated iWD by some other entity (not this Database Capture Point). For example, the task could have been updated by a user through the iWD Global Task List, by another capture point, by a routing strategy, or by an employee desktop application. Note that since there is only a single query that is executed based on all task updates, you must include all task attributes in this query that could possibly be updated.	update cases set casePriority=?'Priority', caseDescr=?'Description' [...] where caseId=?'ExternalId'
resumedUpdateSql	This query updates the corresponding database record to reflect that the associated task has been resumed (i.e. from a Held status).	update cases set status = 'resumed' where caseId=?'ExternalId'

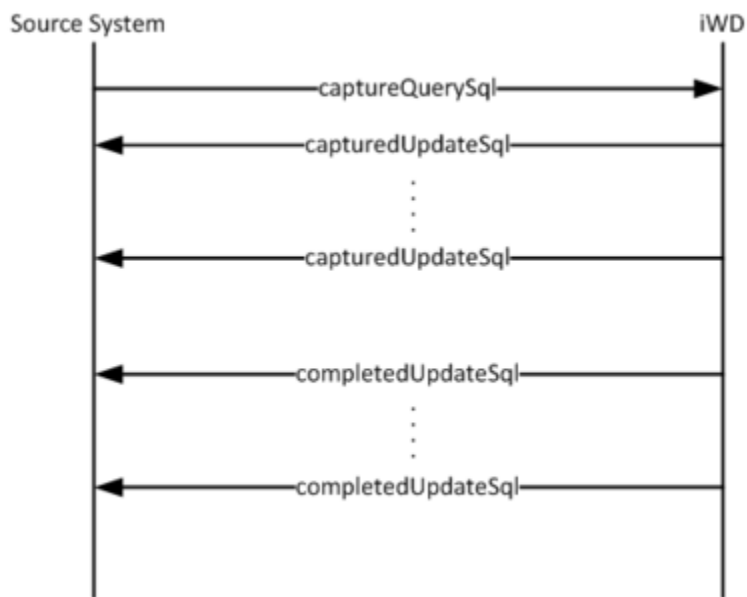
Source Update Queries

Source Update Query	Corresponding iWD Event	Example
sourceUpdateQuerySql	This query returns a set of rows, where each row represents an update request. Each such update request may contain one or more columns that represent iWD task attributes. The name of the column represents the name of the iWD task attribute and the value is the new value of that task attribute. Each row of the result set must	select caseId "ExternalId", updatedTimestamp "SomeTime", casePriority "Priority" from case_updates where status='new'

Source Update Query	Corresponding iWD Event	Example
	contain either InteractionId or ExternalId. If both InteractionId and ExternalId are contained in a row, the value of InteractionId will be used to access the interaction, and the value of ExternalId will be treated as one of the attributes to update.	
sourceUpdatedUpdateSql	This update (or delete) query will execute against a special table in the source database to mark a particular update as having been processed.	update case_updates set status='applied' where caseId=?'ExternalId'
sourceErrorUpdateSql	This update query is executed when there is an error executing an update request (the one that is returned by sourceUpdateQuerySql). Besides the values available from the corresponding capture query, additional values ErrorCode (integer) and ErrorDescription (string up to 256 characters) are available to this query.	update case_updates set status='error', errorcode=?'ErrorCode', errordescr=?'ErrorDescription' where caseId=?'ExternalId'

startupQuerySql

There is one additional query that may be configured, that does not fall into any of the previous three categories. It is called startupQuerySql. This optional query runs once, after the Database Capture Point successfully establishes a connection to the database.



The diagram below provides an example of a simple iWD Database Capture Point flow. The initial captureQuerySql request can return multiple task records. The capturedUpdateSql and completedUpdateSql queries are executed once for each task returned in the captureQuerySql query. More complex system interactions would involve any number of the above queries.

Integration

To use the Database Capture Point, database fields must be added to the source system database tables/views to capture updates from iWD. For example, a field must be added for the capture status as described in the capturedUpdateSql query. This allows iWD to optimize the capturing of new tasks and avoid importing existing tasks repeatedly.

SQL queries must be written that correspond to the capture point configuration parameters described above. Each use case must be tested to ensure that the correct data gets propagated to iWD, and back to the source system database. Batch (.bat) files can be created to quickly execute SQL queries to insert data into database tables on a test database, and other SQL queries to read and output the results.

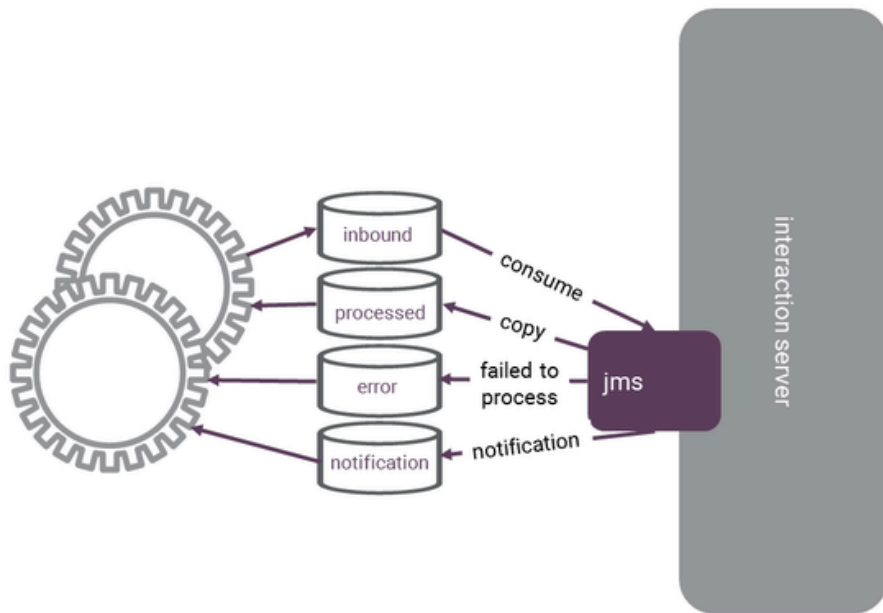
Further Reading

There is more detailed technical reference information for this capture point in the [Database Capture Point](#) section of the eServices Integrated Capture Point Reference Guide.

JMS Capture Point

The JMS Capture Point uses the Java Message Service (JMS) as a way to pass messages between the source system and iWD. JMS is a queue-based message system, and the iWD JMS Capture Point utilizes four queues:

- An inbound queue that the source system will use to submit XML messages. iWD will pick up the XML messages from this queue and use the information in the message to take some action such as creating a new task, completing or updating an existing task, and so on.
- A processed queue where iWD will place a copy of messages that were successfully processed.
- An error queue where iWD will place a copy of messages that were not successfully processed.
- A notification queue where iWD will place both replies to inbound requests, as well as unsolicited notification messages. The notifications received will be based on the notification filtering that is configured for the JMS Capture Point. It is possible, for example, to filter out certain types of notifications coming from iWD that may not be of interest to the source system.



GENESYS

Figure 2: JMS Capture Point Queues

The source system is responsible for generating the XML messages and putting them in the inbound queue on the JMS provider. The JMS Capture Point is an excellent choice for all source systems that can support it, either directly or through some middleware such as WebSphere MQ Server or TIBCO Enterprise Messaging Server. Because it is an enterprise-class message bus, these providers that implement the JMS specification are able to leverage the persistence of the message queues. This

means that if there is some service interruption on either JMS client (that is, the source system or iWD), any messages that have not yet been consumed will remain in the JMS queues to be retrieved when the JMS client is brought back into service. The JMS specification caters for this by using a correlation ID to map the outgoing request message with its corresponding reply.

The JMS Capture Point can also be configured to support SSL between the Genesys Interaction Server and the JMS provider.

Message Transformation

The JMS Capture Point supports optional message transformation. This is provided via the Groovy scripting language. That is, Groovy scripts may be written that describe to iWD how the source system's native XML message format should be converted into a message format that can be understood by iWD, and vice-versa for message replies and notifications coming from iWD back to the source system. For customers who want to use the message transformation feature, Interaction Server must load a JVM, which requires that the Java JDK (latest release of 5 or 6) be installed on the Interaction Server-accessible host.

For more details about message transformation, please read the [Transformation](#) topics in the eServices Integrated Capture Points Guide.

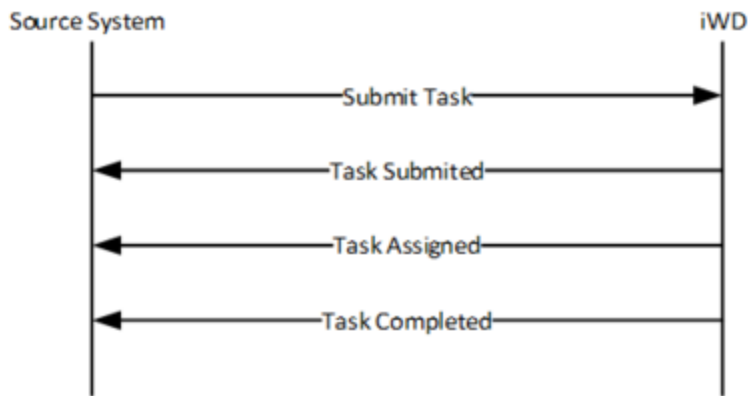
Example JMS Capture Point Request

Here is an example of a simple request using the iWD JMS Capture Point, which instructs iWD to create a new task.

```
<?xml version="1.0" encoding="UTF-8"?>
<interaction operation="submit">
  <properties>
    <ExternalId>SomeExternalId</ExternalId>
    <CustomerSegment>Gold</CustomerSegment>
    <CustomerInfo>
      <FirstName>William</FirstName>
      <LastName>Bell</LastName>
    </CustomerInfo>
  </properties>
</interaction>
```

Simple JMS Capture Point Flow

Here is an example of a simple JMS Capture Point message flow.



Integration

To use the JMS Capture Point, the source system must be able to create XML files and put them into a particular JMS queue. The XML files must contain messages that are in an iWD-compatible (or Interaction Server-native) XML format. If not, then Groovy **transformation** scripts (inbound and outbound) must be created to transform the inbound XML message into the Interaction Server message format, and to transform the outbound reply or notification XML message from Interaction Server into the source system's XML format. There are sample Groovy scripts provided with Interaction Server that can serve as starting points, if message transformation is required.

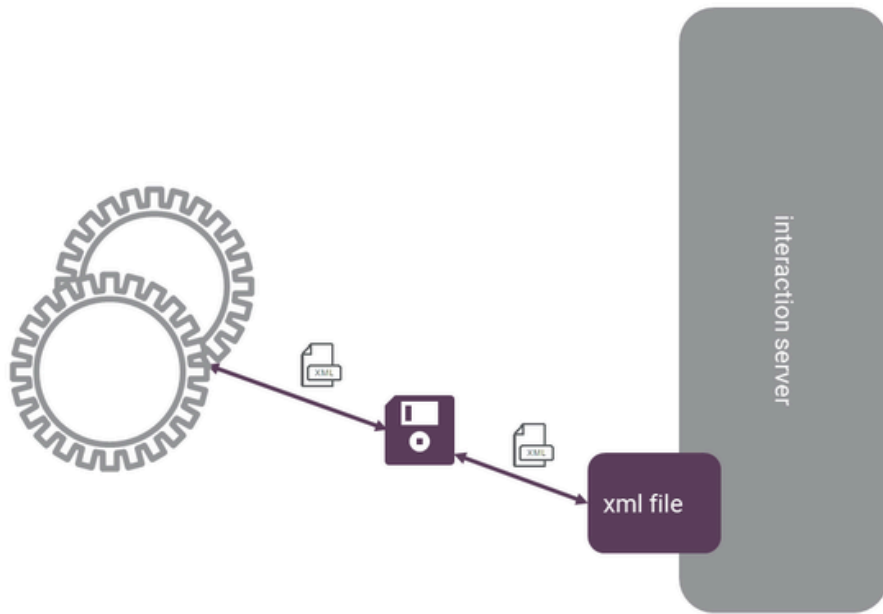
At a minimum, the source system must be integrated with the JMS provider to allow the source system to put the XML messages into the inbound queue. Optionally, the additional JMS queues can be leveraged as part of the iWD integration. The actual configuration of queues on the JMS provider should be trivial; the additional effort will be to read the messages from those queues in order to take some action on those replies or notifications, within the source system's workflow.

Further Reading

There is more detailed technical reference information in the **JMS Capture Point** section of the eServices Integrated Capture Point Guide.

XML File Capture Point

The XML File Capture Point provides the ability to capture and update tasks from XML files that are found in a specified inbound directory on a local or network file system. Additional directories on the file system are used for the capture point to place unsolicited notification messages about the task as it moves through its lifecycle in iWD and changes status, as well as when some updates are made to the task in iWD.



GENESYS

The XML File Capture Point scans the inbound directory based on a configurable time period, which can be as frequently as every 5 seconds.

It is possible to configure the XML File Capture Point to filter out specific types of unsolicited notifications, to reduce the number of messages being produced by the capture point that may not have value to the source system.

An inbound XML file may contain multiple operations—for example, Create Task and Complete Task messages for different Capture IDs, or multiple Create Task messages to create more than one task with a single XML file).

The XML File Capture Point may also be used to process tasks that originate as spreadsheet data. In this case, a script would need to be created to put the spreadsheet data into an XML format to then move it to the appropriate directory on the file system to be consumed by the capture point. Alternatively, an XML data map can be created directly in Excel, using the iWD message schema (xsd), after which the data from the Excel file can be exported as an XML file and put directly into the file system directory.

Supported Directories

The XML File Capture Point uses the following directories on the local or network file system, to process messages:

- **Inbound**—The directory from which the tasks are captured. This directory is also used to process messages coming from the source system such as Complete Task and Update Task.
- **Error**—If a file from the inbound directory is impossible to parse or otherwise process, and no corresponding task has been created, the original file is copied to this directory.
- **Processed**—If a file from the inbound directory has been successfully processed and its corresponding task has been created, the original file is copied into this directory.
- **Notification**—All solicited and unsolicited notifications, resulting from processing of task captured by this capture point will be written in the form of .xml files into this directory, subject to the notification filtering settings.
- **Completed**—If a task is completed in the iWD system, an iWD Task Info notification will be saved in the form of an .xml file into this directory.
- **Rejected**—If a task is put into a rejected state in the iWD system (which is based on the business logic configured in the iWD business process), an iWD Task Info notification will be saved in the form of an .xml file into this directory.
- **ErrorHeld**—If a task is put into an error state in the iWD system (which is based on the business logic configured in the iWD business process), an iWD Task Info notification will be saved in the form of an .xml file into this directory.
- **Canceled**—If a task is canceled in the iWD system, an iWD Task Info notification will be saved in the form of an .xml file into this directory.

Message Transformation

The XML File Capture Point supports optional message transformation. This is provided via the Groovy scripting language. That is, Groovy scripts may be written that describe to iWD how the source system's native XML message format should be converted into a message format that can be understood by iWD, and vice-versa for message responses and notifications coming from iWD back to the source system.

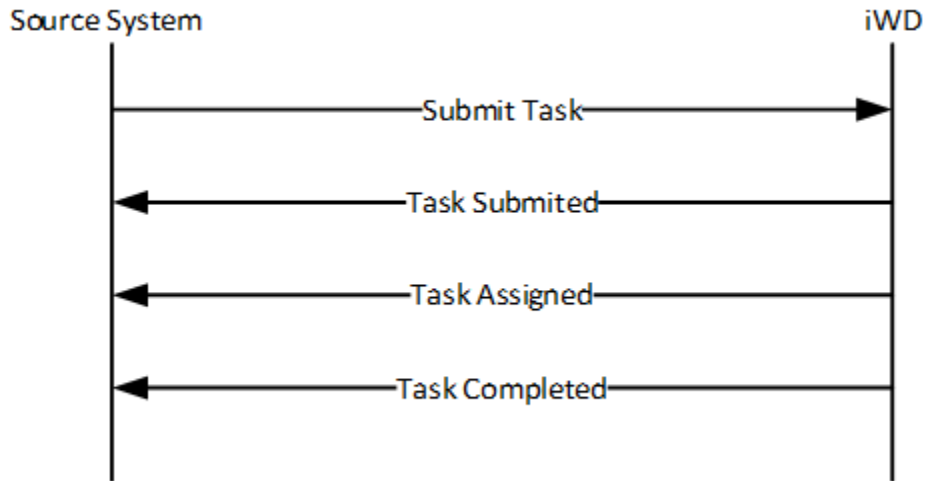
For more details about message transformation, please read the [Transformation](#) topics in the eServices Integrated Capture Points Guide.

Example XML File Capture Point Request

Here is an example of a simple request using the iWD XML File Capture Point, which instructs iWD to create a new task. This request includes examples of iWD core, extended, and custom attributes. The core attributes are those such as captureId, which can be seen near the top of the request. The extended attributes are those that are enclosed within the <ext></ext> elements, and the custom attributes are the key-value pairs that are enclosed within the elements.

```
<?xml version="1.0" encoding="UTF-8"?>
<interaction operation="submit">
  <properties>
    <ExternalId>SomeExternalId</ExternalId>
    <CustomerSegment>Gold</CustomerSegment>
    <CustomerInfo>
      <FirstName>William</FirstName>
      <LastName>Bell</LastName>
    </CustomerInfo>
  </properties>
</interaction>
```

Simple XML File Capture Point Message Flow



Integration

To use the XML File Capture Point, the source system must be able to generate XML files and write/export them to a local or network file system. The XML files must contain messages that are in an iWD-compatible (or Interaction Server-native) XML format. If not, then Groovy transformation scripts (inbound and outbound) must be created to transform the inbound XML message into the Interaction Server message format, and to transform the outbound reply or notification XML message from Interaction Server into the source system's XML format. There are sample Groovy scripts provided with Interaction Server that can serve as starting points, if message transformation is required.

It is easy to test the XML File Capture Point by putting a new message in the inbound directory, seeing if it gets picked up by the capture point, and observing how it is processed. It is possible to include multiple messages in a single XML file (for example, several Create Task messages), allowing you to test multiple use cases with one file.

Further Reading

There is more detailed technical reference information for this capture point in the [XML File Capture Point](#) section of the eServices Integrated Capture Point Reference Guide.

Common Features

This section describes several features that are common to all (or multiple) types of iWD capture points.

Default Attribute Values

For simplicity, it is possible to configure default values for any task attribute. A common example is Media Type, which is a core task attribute that once defined for a task, cannot be changed. Genesys predefines many Media Types such as workitem, webform, and sms, but it is possible to extend this list and add custom media types. If you configure a default value for a task attribute such as Media Type, on a per-capture-point basis, it means that this default value will be used whenever a new task is captured from this capture point. It eliminates the need to include this attribute within every Create Task message. Other attribute values for which it is helpful to set defaults include:

- InteractionType
- InteractionSubtype
- IWD_solutionId
- IWD_tenantId
- IWD_capturePointId
- TenantId
- Queue.

Alternatively, values for these last two attributes may be set by defining an endpoint for the capture point, which indicates the Genesys Tenant and Queue into which all new tasks shall be placed after they are created in Genesys Interaction Server.

Notification Filtering

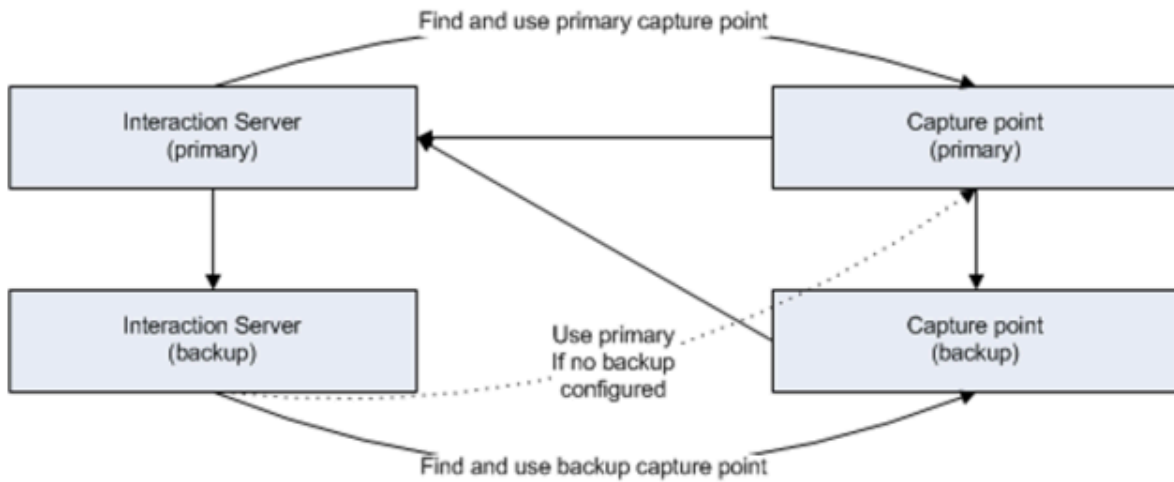
For these capture points that support notifications, it is possible to configure the system to suppress specific types of notifications coming from iWD to the source system:

- JMS
- XML File
- Database

Filtering notifications can help improve overall performance, because the capture points are capable of generating notifications for every update that happens to the task. These updates can be very frequent depending on the nature of the business process.

Redundancy

All the capture points support primary/backup—or warm standby—configuration. The primary Genesys Interaction Server (based on configuration) will search for the primary capture point application and use its configuration to start the capture point. The backup Interaction Server will search for backup capture point application and use its configuration to start the capture point. If there is no backup capture point configured, the backup Interaction Server will use the primary capture point application.



Selecting an Appropriate Capture Point

Selecting the appropriate capture point to use for each source system requires an understanding of the capabilities of the source system. For example, some source systems such as business process management (BPM) systems or systems that use enterprise message middleware such as WebSphere MQ Server or TIBCO Enterprise Message Service, will support the Java Message Service (JMS). Other source systems that include a workflow component will have the ability to integrate with SOAP-based Web services. Older systems that lack these integration capabilities may be able to generate text files. These text files may already be in an XML format but if not, scripting can be used to transform the message from the source system's native text format to an XML format that can be read by iWD. Finally, systems that do not support JMS or Web services, and which cannot produce flat files, will still have operational databases. iWD can directly query the database of such systems in order to create tasks, update tasks, etc.

The capture point you select will depend on the capabilities of the source system from which you are capturing tasks. In some cases, there will be multiple options to select from. Therefore, it is useful to know the capabilities and limitations of the various capture points.

JMS Capture Point

When possible, you should use the **JMS Capture Point** if the JMS (Java Message Service) enterprise messaging service is supported by your source system. This capture point is bi-directional, supporting an input queue and an output queue. Since it uses an enterprise messaging system, it is the most reliable way to set up communication between the source system and iWD. For example, if iWD has a notification to provide to the source system, that notification will be placed in a JMS message queue by the JMS Capture Point. If there is a loss of communication between the JMS Provider and the source system, the notification message will remain in queue until communication is restored.

Database Capture Point

The **Database Capture Point** is also bi-directional and is very flexible. However, to leverage the bi-directionality, it is necessary to update tables on the source system database. In some environments, this will not be possible.

Web Service Capture Point

The **Web Service Capture Point** is an excellent choice for integration with any source system that has the capability of invoking SOAP/HTTP messages from within its workflow. The message set of the Web Service Capture Point is very straightforward and is easy to integrate. The SOAP payload, in XML format, is easy to understand and is logically organized.

While the Web Service Capture Point is not bi-directional, you can use the Web Service Capture Point from the source system to request the latest status for a particular task, including the current values

of all the task attributes. This could be done from the source system just before taking certain actions on the source system, to ensure that the latest updates that might have occurred on the iWD side, can be propagated to the source system.

XML File Capture Point

The **XML File Capture Point** is a good option when it is being integrated with legacy host systems that do not have support for web services or modern messaging systems. In most cases, these systems can generate flat files that consist of lists of attributes for each work item in its database. These flat files could be converted to an XML format and then read by the iWD XML File Capture Point. The XML File Capture Point is bi-directional in that notifications from iWD are put into directories on the local or network file system that can then be picked up and read by the source system.

Bear in mind that the XML File Capture Point involves more I/O with the file system, which may carry a performance overhead.

Message Transformation – Yes or No?

The integrated capture point functionality in Interaction Server supports optional message transformation for the File and JMS capture points. Internally, these capture points work with messages in XML format. XML message transformation can be applied to each incoming and outgoing message, allowing integration with custom interaction definitions and XML formats.

Using XML message transformation with the JMS or XML file capture point decreases the overall throughput of the capture point, so that should be considered. In low-to-mid-volume deployments it may not matter, but for very high-volume deployments Genesys recommends not using XML message transformation. From a practical standpoint this means using the Interaction Server native message format. If the source system cannot create an Interaction Server-compatible XML message, the only way to avoid using capture point message transformation is to transform the message on the source system prior to submitting it to iWD.