



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

Web Service Capture Point

5/8/2025

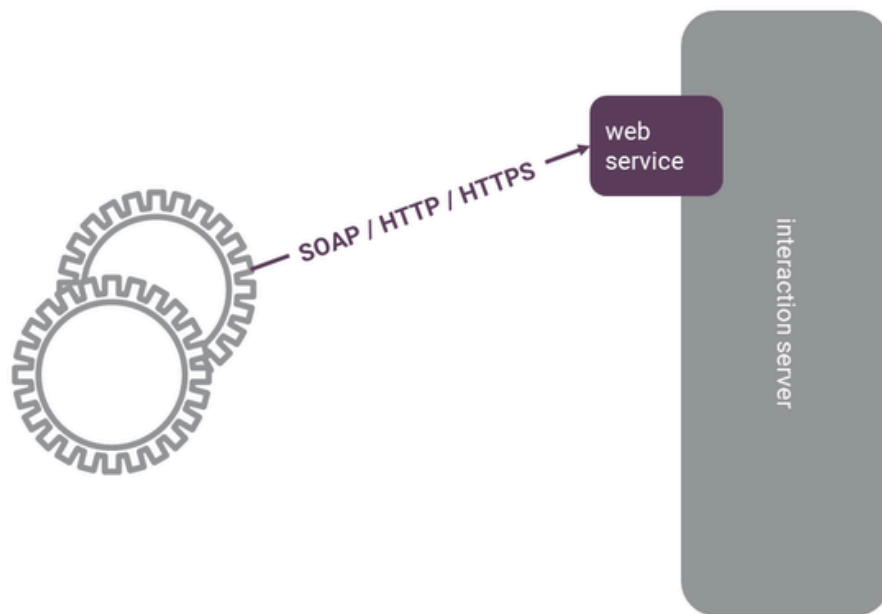
Contents

- 1 Web Service Capture Point
 - 1.1 Example Web Service Capture Point Request
 - 1.2 Integration
 - 1.3 Further Reading

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.