Integrated Capture Points Guide

Transformation

5/8/2025

Contents

Transformation

The integrated capture point functionality in Interaction Server supports optional message transformation for the File, JMS, and Kafka 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.

A sample iWD compatibility transformation script is included with the installation of Interaction Server and can be used as a basis for customization. The sample script is a transformation script that allows you to format an XML message according to the iWD 8.0 schema, and then have it transformed into Interaction Server's native message format.

Inbound vs. Outbound Transformations

The *inbound* transformation Groovy script (if specified by the `xsl-inbound-transform-path` option) is called when a capture point needs to transform an inbound XML message. The *outbound* transformation Groovy script (if specified by the `xsl-outbound-transform-path` option) is called when a capture point needs to transform outbound XML message.

Configuration Options

In order to enable transformation, the following options must be configured in the `settings` section of the Capture Point Application object:

For full descriptions of each of the following options, refer to the *eServices Reference Manual.*

- `xsl-inbound-transform-path`—String representation of a Uniform Resource Identifier (URI) that points to a shared Groovy script file containing the inbound transformation script.
- `xsl-outbound-transform-path`—String representation of a URI that points to a shared Groovy script file containing the transformation scripts for outbound notifications.

Transformers Interface

The interface for the transformation script is defined as follows:

```
package com.genesyslab.eservices.interactionserver.capturepoints.xmltransformer;
public interface XmlTransformer
{
        void init(java.util.Properties parameters);
        byte[] transform(byte[] inputXml, java.util.Properties parameters);
        void cleanup();
        void setLogger(Logger logger);
        void reconfigure(java.util.Properties parameters);
}
```

Any custom script should implement this interface in order to be usable by Interaction Server. For a good starting point for a custom script, refer to the sample scripts that are provided with Interaction Server.

When Interaction Server creates a transformer, it calls its `init` method and passes all of the parameters that are defined in the `inbound-transformer-parameters` section (for inbound transformation scripts) or in the `outbound-transformer-parameters` section (for outbound transformation scripts). The transformer object should store these properties for possible future use during transformation.

The main functional method `transform` transforms the `inputXML` XML message into the required form and returns the transformed XML message that will be either parsed by Interaction Server (for inbound messages) or, for outbound messages, put directly into notification message queue (for message queue capture points). Each call to the `transform` method by Interaction Server can be given a set of properties to account for during the individual transformation process of a single document. Interaction Server provides the following parameters to the transformation method:

Parameter	JMS capture point	File capture point	Kafka capture point
CapturePointName	The name of the capture point invoking the transformation	The name of the capture point invoking the transformation	The name of the capture point invoking the transformation
CurrentTime	The current UTC timestamp in the format `YYYY-MM-DDTHH:MM:SSZ`	The current UTC timestamp in the format `YYYY-MM-DDTHH:MM:SSZ`	The current UTC timestamp in the format `YYYY-MM-DDTHH:MM:SSZ`
`CapturePointType` (Only for outbound transformation)	The type of capture point - jms	The type of capture point - file	The type of capture point - kafka
`MessageId` (Only for inbound transformation)	The `JMSMessageId` property of the JMS message being transformed	The file name of the inbound file currently being transformed	The artificial unique identifier consisting of partition number the message was consumed from and its offset divided by dot (.)
`CorrelationId` (Only for outbound transformation)	The `JMSCorrelationId` property of the JMS message being transformed		The correlation ID as it was read from the inbound message header as specified in the `correlation-id-header-key` option
`OutboundMessage`	Contents of the original outbound JMS message		
`InboundMessage`	Contents of the original inbound JMS message		
`CopyOriginalHeaders`			`true` or `false` as specified in the `copy-original-headers-in-reply` option
`InboundRecord` (Only for inbound transformation)			The ConsumerRecord object which the message being transformed was received from.
`OutboundHeaders`			The List object in which Kafka message headers to be included

Parameter	JMS capture point	File capture point	Kafka capture point
(Only for outbound transformation)			into an outbound message can be added during transformation.

The `Logger` interface provided to the script allows for logging of any diagnostic or error messages into the Interaction Server log by the same means that Interaction Server logs messages. Logging configuration works the same as for any other Interaction Server messages, including logging to the console, a file, or network logging. The `Logger` interface is defined as follows:

```
package com.genesyslab.eservices.interactionserver.capturepoints.xmltransformer;
public interface Logger
{
    public static enum LogLevel { DEBUG, TRACE, STANDARD };
    public void log(LogLevel level, String logMessage);
}
```

XML Encoding Considerations

Interaction Server can parse XML messages in the following encodings:

- UTF-8
- UTF-16
- ISO-8859-1
- US-ASCII

Outbound notification messages are encoded in UTF-8. This requires the transformation scripts to provide output in one of the supported encodings (for inbound transformations) and to be capable of parsing the UTF-8 encoded XML messages (for outbound transformation scripts).

Because Groovy scripts use the `XmlParser` class to parse XML messages, they have no difficulty processing UTF-8 and can support any encoding supported by Java (depending on the installed packages). The outbound transformation scripts can also produce outbound XML messages in any encoding if the appropriate Java packages are installed. The outbound transformer provided with Interaction Server generates output in UTF-8 and is capable of generating output (without any additional Java packages) in the following encodings:

- US-ASCII
- ISO-8859-1
- UTF-8
- UTF-16BE
- UTF-16LE
- UTF-16

Care should be taken to correctly handle encoding in Groovy. The recommended place to look for an example is the transformation scripts that are provided with the Interaction Server installation. The following pattern shows how to correctly generate XML in the required encoding and with appropriate XML declaration:

```
def outputDoc = new StreamingMarkupBuilder()
outputDoc.encoding = "utf-8"
def outputStream = new ByteArrayOutputStream()
new OutputStreamWriter(outputStream, outputDoc.encoding) << outputDoc.bind {
        mkp.xmlDeclaration()
        somecontent {
    }
}
return outputStream.toByteArray()
```

iWD Compatibility Transformation Scripts

There are two Groovy scripts provided for transformation of inbound and outbound messages to and from iWD message format. These scripts provide backward compatibility with iWD 8.0 message format considering the structure of the iWD specific business process provided with iWD 8.0. The iWD 8.0 message format is described in detail in the iWD 8.0 Deployment Guide. Only general rules of the transformation process are described here. The provided transformation scripts below are only for standard iWD messages as specified in the document. For custom messages, customization of these scripts is necessary.

- Inbound Transformation Script
- Outbound Transformation Script