TIBCO Foresight® REST API

User Guide

Version 1.2.0 | September 2024

Contents

Introduction

The TIBCO Foresight® REST API is a front-end API to the TIBCO Foresight® Instream® and the TIBCO Foresight® Translator engines. Foresight® REST API is a Java-based application and uses the Foresight® Instream and Foresight® Translator Java libraries for validation and translation processing.

Validation Process

TIBCO Foresight® Instream® validates an EDI datafile against a guideline and reports any errors found. The user can also produce EDI response documents if the Response Generator options are specified, and can split the datafile into smaller parts if the Doc Splitter options are specified.

Inputs	Outputs
EDI datafile	Validation results
Guideline	Response documents (Optional)
Validation controlfile (Optional)	Split files (Optional)
Response Generator controlfile (Optional)	
Doc Splitter controlfile (Optional)	

Translation Process

Foresight Translator translates an EDI datafile into other formats, such as XML, Flatfile, EDI, etc. and vice versa. To perform the translation, it uses two guidelines or schemas (a ‘from’ guideline/schema and a ‘to’ guideline/schema) and a map that links the from fields to the to fields.

Inputs	Outputs
EDI or XML data file or Flatfile	Translated data files
Map file	
Source (or From) guideline or schema	
Target (or To) guideline or schema	
Translation type	

Technical Overview

Foresight REST API is based on Java API/DLLs and uses four REpresentational State Transfer (REST) operations: PUT, GET, POST, and DELETE. It uses the following resources as input, for processing and for generating the output:

- Guidelines (PUT, GET, and DELETE)
- Data Files (PUT, GET, and DELETE)
- Control Files (PUT, GET, and DELETE)
- Map Files (PUT, GET, and DELETE)
- Jobs (POST, GET, and DELETE)
- Output Files (GET, and DELETE)

Foresight REST API includes four types of callbacks:

- Foresight Instream/Foresight TranslatorMid-Processing Callback
- File Request Callback
- Notification Callback
- Message Callback

Input Resources

Input Resources use the PUT, GET, and DELETE operations. The validation and the translation processes function in different manners.

In the validation process, the input includes an EDI datafile and a guideline. It requires several other optional files.

In the translation process, the input includes an EDI or XML datafile and a map file. In addition, it must have a source and target schema or guidelines.

The Foresight REST API server requires the following Input Resources:

- Guidelines
- Datafiles
- Controlfiles
- Maps

Guidelines

A guideline resource is either represented as a SEF (.sef or .std extension) or an XML schema (.xsd extension). If an extension is not specified in the Foresight REST API command, the system first looks for a file with a .std extension. If not found, it searches for a file with a .sef extension.

Three Foresight REST API operations are supported for guideline resources: PUT, GET, and DELETE.

Using the PUT operation

The user can utilize the PUT operation to upload guidelines to the Foresight REST API server for use in a subsequent job operation. All guidelines uploaded through the PUT operation go into the Database folder under the Foresight REST API server's guidelines directory.

1. REST Path

   `.../guidelines/guidelineFileName`

   `.../guidelines/zip/zipFileName`

2. Optional Parameters

   overwrite=Y

   If a guidelineFileName exists on the server, the server rejects the request unless this optional parameter is specified.

3. Request Body: guideline file (Media Type=application/octet-stream)

   Note: To upload multiple guidelines, provide a zip file as an input.

4. Possible Responses

Response	Description
200 OK	The guideline upload is successful.
400 Bad Request	Problem with the request (missing guideline ID) or with the guideline that is sent (does not appear to be a .sef file); see the response body for the reason.
409 Conflict	The guideline cannot be overwritten; see the response body for additional information.
415 Unsupported Media Type	The guideline PUT supports a media type of ‘application/octet-stream’ only.
500 Internal Server Error	The server had problems writing files; see the response body for the reason.

5. Examples

   a. http://<host-name>:<port>/ForesightREST/guidelines/TEST835.std

   b. http://SERVX:8080/ForesightREST/guidelines/TEST835.std?overwrite=Y

Using the GET operation

The GET operation is used to

- Get the information about a specific guideline file on the Foresight REST API server, or
- Get a list of all the guidelines on the Foresight REST API server.

For specific guideline GETs, the following information is returned in the response body:


```
{
    "stdFileInfo": "20220228 072106.000+0530/139350",
    "internalId": "834AA120",
    "stdDate": "07/27/110:15:56:54",
    "size": "139350",
    "guidelinePath":
"\\usr\\local\\tomcat\\TibcoFSRest\\guidelines\\Database\\834AA120.std",
    "id": "834AA120",
    "type": "X12",
    "desc": "Types 1-2,  Addenda October, 2002 Included;  Benefit
Enrollment and Maintenance (834-A1)",
    "fileMod": "20220228 072106.000+0530"
}
```


1. REST Path
    a. For a specific guideline

       `.../guidelines/guidelineFileName`

    b. For a list of all the guidelines

       `.../guidelines`

    c. For bytecode of zip file of all the guidelines

       `../guidelines?multiple=Y`

2. Optional Parameters
    a. full=Y

       For a specific guideline, where the guideline body is desired and installation permits it, this option causes the server to return the .std/.sef file contents in the response body (media type=Application/Octet-Stream). If requesting a list of all the guidelines, this parameter is ignored.

    b. multiple=Y

To get the bytecode of the zip file of all the guidelines.

3. Request Body: (Empty)
4. Possible Responses

Response	Description
200 OK	The guideline information retrieval (or full guideline retrieval) is successful; response body contains the information. For full retrieval, the body contains the contents of the guideline requested from the server. For a normal GET, the body contains the information about the guideline file on the server.
400 Bad Request	Problem with the request (such as missing guideline id); see the response body for reason.
403 Forbidden	The specified guideline is a 'Factory' guideline and cannot be downloaded.
404 Not Found	The specified guidelines cannot be found on the server.
415 Unsupported Media Type	The guideline GET supports media types of 'application/octet-stream' and 'text/plain'.
500 Internal Server Error	The server had problems reading the file; see the response body for reason.

5. Examples
    a. http://<host-name>:<port>/ForesightREST/guidelines/PDSA837I

    Response Body:

```
{
    "stdFileInfo": "20201022 081648.000-0500/755533",
    "internalId": "5010-837I_TRANS",
    "stdDate": "08/26/109:15:34:38",
    "size": "755533",
    "guidelinePath": "c:\\FSREST\\guidelines\\Database\\5010-
```

```
837I_TRANS.std",
    "id": "5010-837I_TRANS",
    "type": "X12",
    "desc": "Types 1-2, Health Care Claim: Institutional
(837I), Version 5, Release 1, Errata (OCT 2007)",
    "fileMod": "20201022 081648.000-0500"
}
```

b. http://SERVX:8080/ForesightREST/guidelines

Response Body:

```
PDSA837I.std 210-X12-4030
269X187Q
269X187R
270-A120
270-X279
270AA120
271-A120
271-X279
271AA120
275-X151
275-X210
275-X314
…
X12-6030 (Factory) X12-6031 (Factory) X12-6032 (Factory) X12-
6040 (Factory) X12-6041 (Factory) X12-6042 (Factory) X12ICS
(Factory) XHL7
XML_PO_F
```

Note: While retrieving files having '?' in the name, the '?' must be replaced by '%3F'.

Using the DELETE operation

The DELETE operation is used to remove a guideline resource from the Foresight REST API server, if permitted by the installation.

1. REST Path: `.../guidelines/guidelineFileName`
2. Request Body: (Empty)
3. Possible Responses

Response	Description
200 OK	The guideline is deleted successfully.
400 Bad Request	Problem with request (missing guideline id, and so on.); see the response body for reason.
409 Conflict	The specified guideline resources cannot be deleted; see the response body for additional information.
500 Internal Server Error	The server had problems writing the files; see the response body for the reason.

Datafiles

A datafile resource is a data file to be validated. Three Foresight REST API operations are supported for datafile resources: PUT, GET, and DELETE

Using the PUT operation

The PUT operation is used to upload a data file to the Foresight REST API server for use in a subsequent job operation. All data files uploaded through this PUT operation go into the Foresight REST API server's datafile directory.

1. REST Path

   `.../datafiles/dataFileName`

   `.../datafiles/zip/zipFileName`

2. Optional Parameters

   a. overwrite=Y

      If a data file being datafilename exists on the server, the server rejects the request unless this optional parameter is specified.

3. Request Body: data file (Media Type=application/octet-stream)
4. Possible Responses

Response	Description
200 OK	The data file upload is successful.
400 Bad Request	Problem with the request (missing data file name, and so on.); see the response body for the reason.
409 Conflict	The data files cannot be overwritten; see the response body for additional information.
415 Unsupported Media Type	The data file PUT supports a media type of ‘application/octet-stream’ only.
500 Internal Server Error	The server had problems writing files; see the response body for the reason.

5. Examples
    a. http://<host-name>:<port>/ForesightREST/datafiles/TESTData.txt
    b. http://SERVX:8080/ForesightREST/datafiles/FlatFile.dat?overwrite=Y

Using the GET operation

You can use the GET operation to:

- Get the information about (or the contents of) a specific data file on the Foresight REST API server, or
- Get a list of all the datafiles on the Foresight REST API server.

For specific data file GET, the following information is returned in the response body:

- File name
- File size
- File creation date and time
- File last modification date and time

Note: The information is in a single line and the fields are tab delimited. When a list of all the control files is requested, multiple lines of the same information are returned, one line per file.

1. REST Path
    a. For a specific data file

       `.../datafiles/dataFileName`

    b. For a list of all the datafiles

       `.../datafiles`

    c. For bytecode of zip file of all the datafiles

       `../datafiles?multiple=Y`

2. Optional Parameters
    a. full=Y

       If requesting a specific data file, and the actual contents of the data file is desired, and the installation permits it, then specifying this option causes the server to return the contents of the specified data file in the response body (media type=Application/Octet-Stream). If requesting a list of all the datafiles, this parameter is ignored.

    b. multiple=Y

       To get the bytecode of the zip file of all the datafiles.

3. Request Body: (Empty)
4. Possible Responses

Response	Description
200 OK	The data file information retrieval (or full data file retrieval) is successful; response body contains the information. For full retrieval, the body contains the contents of the datafile requested from the server. For a normal GET, the body contains the information about the data file on the server.

Response	Description
400 Bad Request	Problem with the request (missing data file name, and so on); see the response body for the reason.
404 Not Found	The specified data files cannot be found on the server.
415 Unsupported Media Type	The data file GET supports media types of ‘application/octet-stream’ and ‘text/plain’.
500 Internal Server Error	The server had problems reading the file; see the response body for the reason.

5. Examples
    a. http://<host-name>:<port>/ForesightREST/datafiles/837I_4010_H_6claims.txt

    Response Body:

```
837I_4010_H_6claims.txt    7179 2019-05-22T21:13:30.214054Z
2017-07-25T23:42:52Z
```

    b. http://SERVX:8080/ForesightREST/datafiles

    Response Body:

```
[
    {
        "fileName": "1_D3_CONTRL.dat",
        "size": "0",
        "created": "2023-10-18T18:17:26.8185517Z",
        "modified": "2021-10-06T21:48:49.3641381Z"
    },
    {
        "fileName": "1_D3_CONTRL_updated.dat",
        "size": "386",
        "created": "2023-10-18T18:17:26.8315535Z",
        "modified": "2021-03-01T17:51:48.3834129Z"
    },
    {
        "fileName": "270_4010_H_3InfoSrc_Clean.txt",
        "size": "3007",
```

```
        "created": "2023-10-18T18:17:26.8585517Z",
        "modified": "2023-12-14T22:29:18.1016401Z"
    },
    {
        "fileName": "270_5010_H_BadReceiverID.txt",
        "size": "611",
        "created": "2023-10-18T18:17:26.8705518Z",
        "modified": "2023-12-14T22:29:18.1106405Z"
    },
    {
        "fileName": "271_4010_H_2InfoSrc_Clean.txt",
        "size": "2060",
        "created": "2023-10-18T18:17:26.8835588Z",
        "modified": "2023-12-14T22:29:18.1156392Z"
    },
    {
        "fileName": "2Interchanges837i.txt",
        "size": "4042",
        "created": "2023-10-18T18:17:26.8455512Z",
        "modified": "2023-09-13T22:38:40.8527996Z"
    },
    {
        "fileName": "bkup",
        "size": "24576",
        "created": "2023-12-18T03:06:42.1330487Z",
        "modified": "2023-12-18T03:06:46.0144448Z"
    }
]
```


Note: While retrieving files having '?' in the name, the '?' must be replaced by '%3F'.

Using the DELETE operation

The DELETE operation is used to remove a data file resource from the Foresight REST API server, if permitted by the installation.

1. REST Path: `.../datafiles/dataFileName`
2. Request Body: (Empty)
3. Possible Responses

Response	Description
200 OK	The specified data file is deleted successfully.
400 Bad Request	Problem with request (missing data file name, and so on); see the response body for the reason.
409 Conflict	The specified data file resources cannot be deleted; see the response body for additional information.
500 Internal Server Error	The server had problems deleting the file; see the response body for the reason.

Controlfiles

A controlfile resource is a supporting file used by the Foresight Instream processes. Some examples include custom .apf files, Response Generator, custom report template files, and so on.

Three Foresight REST API operations are supported for controlfiles resources: PUT, GET, and DELETE.

Using the PUT operation

The PUT operation is used to upload a control file to the Foresight REST API server for use in a subsequent Job operation. All control files uploaded through this PUT operation go into the Foresight REST API server's controlfile directory.

1. REST Path

   `.../controlfiles/controlFileName`

   `.../controlfiles/zip/zipFileName`

3. Optional Parameters

   overwrite=Y

   If a control file being 'PUT' exists on the server, the server rejects the request unless this optional parameter is specified.

4. Request Body: controlfile (Media Type=application/octet-stream)

> **Note:** To upload multiple guidelines, provide a zip file as an input.

5. Possible Responses

Response	Description
200 OK	The control file upload is successful.
400 Bad Request	Problem with request (missing control file name, and so on); see the response body for the reason.
409 Conflict	The control file cannot be overwritten; see the response body for additional information.
415 Unsupported Media Type	The control file PUT supports a media type of ‘application/octet-stream’ and ‘text/plain’.
500 Internal Server Error	The server had problems writing a file; see the response body for the reason.

5. Examples
    a. http://<host-name>:<port>/ForesightREST/controlfiles/TESTErrs.apf
    b. http://SERVX:8080/ForesightREST/controlfiles/MyCustRespTemplate.txt?overwrite=Y

Using the GET operation

You can use GET operation to:

- Get the information about a specific control file on the Foresight REST API server,
- Get a list of all the control files on the Foresight REST API server.

For specific data file GETs, the following information is returned in the response body:

- File name
- File size
- File creation date and time

- File last modification date and time

Note: The information is in a single line and the fields are tab delimited. When a list of all the control files is requested, multiple lines of the same information are returned, one line per file.

1. REST Path
    a. For a specific control file

    ```
    .../controlfiles/controlFileName
    ```

    b. For a list of all control files

    ```
    .../controlfiles
    ```

    c. For bytecode of zip file of all the controlfiles

    ```
    ../controlfiles?multiple=Y
    ```

2. Optional Parameters
    a. full=Y

    If requesting a specific control file, and the actual contents of the control file is desired, and the installation permits it, then specifying this option causes the server to return the contents of the specified control file in the response body (media type=Application/Octet-Stream). If requesting a list of all the control files, this parameter is ignored.

    b. multiple=Y

    To get the bytecode of the zip file of all the controlfiles.

3. Request Body: (Empty)
4. Possible Responses

Response	Description
200 OK	The control file information retrieval is successful; the response body contains the information. For full retrieval, the body contains the contents of controlfile requested from the server. For a normal GET, the body contains the information about the control file on the server.

Response	Description
400 Bad Request	Problem with request (missing control file name, and so on); see the response body for the reason.
404 Not Found	The specified control files cannot be found on the server
415 Unsupported Media Type	The control file GET supports media types of ‘application/octet-stream’ and ‘text/plain’.
500 Internal Server Error	The server had problems reading the file; see the response body for the reason.

5. Examples
    a. http://<host-name>:<port>/ForesightREST/controlfiles/Phase2.apf

    Response Body:

    ```
    Phase2.apf    6779 2019-05-10T23:41:03.629214Z 2019-05-
    11T03:09:22Z
    ```

    b. http://SERVX:8080/ForesightREST/controlfiles

    Response Body:

RGtemplate837I_c.txt	337 2019-05-22T21:13:30.192058Z 2017-07-25T23:42:52Z
test.apf	8416 2019-05-08T19:30:56.601053Z 2019-05-08T19:36:06.433946Z
test2.apf	9416 2019-05-08T19:33:48.212871Z 2019-05-08T19:36:06.433946Z

Note: While retrieving files having '?' in the name, the '?' must be replaced by '%3F'.

Using the DELETE operation

The DELETE operation is used to remove a controlfile resource from the Foresight REST API server, if permitted by the installation.

1. REST Path: `.../controlfiles/controlFileName`
2. Request Body: (Empty)
3. Possible Responses

Response	Description
200 OK	The specified control file is deleted successfully.
400 Bad Request	Problem with the request (missing control file name, and so on); see the response body for the reason.
409 Conflict	The specified control file resources cannot be deleted; see the response body for additional information.
500 Internal Server Error	The server had problems deleting the file; see the response body for the reason.

Maps

A map resource is a translation map file used by the Foresight Translator processes. A map resource connects the fields between different formats.

Three Foresight REST API operations are supported for mapping resources: PUT, GET, and DELETE

Using the PUT operation

The PUT operation is used to upload a map file to the Foresight REST API server for use in a subsequent job translation operation. All map files uploaded through this PUT operation go into the Foresight REST API server's maps directory.

1. REST Path

```
.../maps/mapFileName
.../maps/zip/zipFileName
```

3. Optional Parameters

   overwrite=Y

   If a map file being 'PUT' exists on the server, the server rejects the request unless this optional parameter is specified.

4. Request Body: map file (Media Type=application/octet-stream)

   Note: To upload multiple guidelines, provide a zip file as an input.

5. Possible Responses

Response	Description
200 OK	The map file upload is successful.
400 Bad Request	Problem with request (missing map file name, and so on); see the response body for the reason.
409 Conflict	The map file cannot be overwritten; see the response body for additional information.
415 Unsupported Media Type	Maps PUT supports a media type of 'application/octet-stream' and 'text/plain'.
500 Internal Server Error	The server had problems writing files; see the response body for the reason.

5. Examples

   a. http://<host-name>:<port>/ForesightREST/maps/837EtoX.map

   b. http://SERVX:8080/ForesightREST/maps/810XMLtoEDI.map? overwrite=Y

Using the GET operation

You can use GET operation to:

- Get the information about a specific map file on the Foresight REST API server, or
- Get a list of all the map files on the Foresight REST API server.

For specific data file GETs, the following information is returned in the response body:

- ID = File name
- mapPath = Full path to map file
- fileInfo = File Last Modification Date and Time/Size

When a list of all the map files is requested, map file names are returned one line per file.

1. REST Path
    a. For a specific map file

       `.../maps/mapFileName`

    b. For a list of all the map files

       `.../maps`

    c. For bytecode of zip file of all the maps

       `../maps?multiple=Y`

2. Optional Parameters
    a. full=Y

       If requesting a specific map file, and the actual contents of the map file is desired, and the installation permits it, then specifying this option causes the server to return the contents of the specified map file in the response body (media type=Application/Octet-Stream). If requesting a list of all the map files, this parameter is ignored.

    b. multiple=Y

       To get the bytecode of the zip file of all the maps.

3. Request Body: (Empty)
4. Possible Responses

Response	Description
200 OK	The map file information retrieval (or full map file retrieval) is successful; response body contains the information. For full retrieval, the body contains the contents of the map file requested from the server. For a normal GET, the body contains the information about the map file on the server.
400 Bad Request	Problem with request (missing map file name, and so on); see response body for the reason.
404 Not Found	The specified map files cannot be found on Server.
415 Unsupported Media Type	The map file GET supports media types of ‘application/octet-stream’ and ‘text/plain’.
500 Internal Server Error	The server had problems reading the file; see the response body for the reason.

5. Examples
    a. http://<host-name>:<port>/ForesightREST/maps/Phase2.apf (application/json)

    Response Body:

    ```
    {
    "fileInfo": "20230403 090358.658-0400/244966",
    "id": "origMapFile",
    "mapPath": "C:\\FSRest\\maps\\origMapFile.map"
    }
    ```

    b. http://SERVX:8080/ForesightREST/maps (text/plain)

    Response Body:

    ```
    837-X222_837_EX.map
    837-X223_EX.map
    837-X223_XE.map
    850_4010_02_GG_DEMO_850_EX.map origMapFile.map
    ```

Note: While retrieving files having '?' in the name, the '?' must be replaced by '%3F'.

Using the DELETE operation

The DELETE operation is used to remove a map file resource from the Foresight REST API server, if permitted by the installation.

1. REST Path: `.../maps/mapFileName`
2. Request Body: (Empty)
3. Possible Responses

Response	Description
200 OK	The specified map file is deleted successfully.
400 Bad Request	Problem with request (missing map file name, and so on); see response body for the reason.
409 Conflict	The specified map file resources cannot be deleted; see the response body for additional information.
500 Internal Server Error	The server had problems deleting the file; see the response body for the reason.

Processing and Output Resources

The processing of the resources and generating output involves various JOB operations. The PUT, GET, and DELETE operations support the processing of resources. The GET and DELETE operations support output generation.

Jobs

A job resource is required to perform any operation. The POST operation creates a job resource on the Foresight REST API with a body that contains the following information necessary to define the work to be done.

- Guideline ID
- Data file name
- Any special controlfiles needed
- Any response documents desired

  The response from a job POST is the job ID assigned to the job. This Job ID helps to obtain the job's status, along with any output files generated.

  Three operations are supported for job resources: POST, GET, and DELETE

POST

The POST operation is used to convey a message to the Foresight REST API server to initiate a job to perform some work. The body of the POST request contains the parameters of the work to be done.

1. REST Path

   `.../jobs`

2. When this parameter is included in the POST request, the results detail file (dtl) is copied to the shared file folder.
3. Request Body: (Media Type = text/plain)

The request body contains lines of job parameters in the format key=value. The various parameters are:

Parameter Key	Area	Description	Example	Corresponding Java API Member
JobID	All	Specify the Foresight REST API Job ID to use for the job instead of an automatic job ID. For details, see JobID.	JobId=Test8	
Mode	All	Sets the processing mode for the job. For details, see Mode . The mode must be one of the following: **ASYNC**: Asynchronous operation (Default) **ASYNCN**: Asynchronous operation with notification on completion **ASYNCO**: Asynchronous operation with	Mode=SYNCO	

Parameter Key	Area	Description	Example	Corresponding Java API Member
		notification, including contents of Response Files and Translated Files on completion **SYNC**: Synchronous operation **SYNCO**: Synchronous operation, including Response Files and Translated Files contents on completion.		
Ops (Required)	All	Main Job operations and sequence. For details, see Ops. The Ops must be one of the following: **I** = Instream **T** = Translation **IT** = Instream then Translation	Ops=IT (for inbound operation: Validate then translate)	

Parameter Key	Area	Description	Example	Corresponding Java API Member
		TI = Translation then Instream **Note:** When a service request is submitted to REST API to invoke Instream and Translator, and only one product is installed, REST API only runs the installed product.		
DatafileRef	All	The name of the data file to use. **Note:** DataFileRef is the same as InDataFileRef from the old Translator interface.	Example: DatafileRef=ProdMedAData.edi	InputFile (Instream), InputFile (Translator)
Data	All	Includes the data to be	“Data”: “ISA*...~IEA*1*0003”	

Parameter Key	Area	Description	Example	Corresponding Java API Member
		validated or translated as part of the Job POST body. For details, see Data.		
CallbackTag	All	Sets the value for the 'tag' field in guideline callbacks from the Foresight Instream and the Foresight Translator engines. (See, ICallbackTag, and TCallbackTag parameters, to set each engine's tag value individually.)	Callback="inbound"	
senderName	All	Specify the sender name to use in any AuditSafe messages generated by Job.	senderName="ABC Trucking"	

Parameter Key	Area	Description	Example	Corresponding Java API Member
receiverName	All	Specify the receiver name to use in any AuditSafe messages generated by Job.	receiverName=”Woody’s Warehouse”	
Debug	All	Debug indicator to add additional logging to the Job. Setting to ‘Y’ causes a list of log messages to be attached to the job. These can be accessed as part of the job status information returned from a Job ‘GET’ operation.	Debug=Y	
GuidelineRef	Instream/ Validation	Name of the guideline file to use.	Example: GuidelineRef=X12-5010	guideline
SGuidelineRef	Translator	Name of the guideline file to use for the source (from)	Example: SGuidelineRef=MYX12.std	

Parameter Key	Area	Description	Example	Corresponding Java API Member
		data. For more details, see Guidelines.		
TGuidelineRef	Translator	Name of the guideline file to use for the target (to) data. For more details, see Guidelines.	Example: TGuidelineRef=MYXML.xsd	
Profile	Instream/ Validation	Name of profile (apf) file to use. For details, see Controlfiles.	Example: Profile=MedPat1.apf	profileOptions
ValReportFormat	Instream/ Validation	Type of validation report format: XML or X = XML report TEXT or T = Text report (default)	Example: ValReportFormat=XML	InStreamOutputXmlFormat
OrigFileInfo	Instream/ Validation	Original File Information in format 'mm/dd/yy hh:mm:ss nnnn path' where ‘mm/dd/yy	OrigFileInfo=02/14/19 14:55:34 2032 C:/HVInStream/Demo Data /TestData.txt	origFileInfo

Parameter Key	Area	Description	Example	Corresponding Java API Member
		hh:mm:ss' is the file creation date and time, 'nnnn' is the file size, and 'path' is the original file path.		
Separators	Instream/ Validation	String representing the segment terminator, element delimiter, and composite subelement delimiter characters. Three formats are allowed: a. ASCII decimal values separated by commas b. ASCII hexadecimal values separated by commas	Examples: Separators=~!" Separators=29,30,31 Separators=0x1E, 0x1F, 0x1D	separator, avoidENV

Parameter Key	Area	Description	Example	Corresponding Java API Member
		c. A string of three characters		
EnvIC	Instream/ Validation	Interchange segment to use for validation. Must be used with EnvFG.	Example: EnvIC=ISA*00* *00* *01*9012345720000 *01*9088877320000 *120225*0856*+*00501*0 00002491*0*T*:~	FSDOCUMENTON LY, envlopISA
EnvFG	Instream/ Validation	Functional Group segment to use for validation. Must be used with EnvIC.	Example: EnvGS=GS*HS*90123457 2000*908887732000*2012 0225*1615*2491*X*0050 10X279A1~	FSDOCUMENTON LY, envGS
StopOnEnvErr	Instream/ Validation	Stops the Foresight Instream validation process if enveloping errors are encountered during		

Parameter Key	Area	Description	Example	Corresponding Java API Member
		validation.		
ICallbackTag	Instream/ Validation	Sets the value for the 'tag' field in guideline callbacks from the Foresight Instream engine.	ICallback="inbound"	
TCallbackTag	Translator	Sets the value for the 'tag' field in guideline callbacks from the Translator engine.	TCallback="outbound"	
TThreadLock	Translator	Switches the Translator processing thread locking on or off for each job. Possible values: Y, YES, or 1: Turns on thread locking N, NO, or 0: Turns off thread locking	TThreadLock=Y	

Parameter Key	Area	Description	Example	Corresponding Java API Member
		Default: NO		
RGDocs	Instream/ Response Gen	Response documents desired. One or more of the following codes separated by commas: 997 = X12 997 response 999 = X12 999 response 824 = X12 824 response 277 = X12 277 response TA1 = X12 TA1 response CONTRL = EDIFACT CONTRL response TEXT = Custom text response (see RGTextTemplateFileRef)	Example: RGDocs=999,277,TA1	RG_Rep997, RG_Rep999, RG_Rep824, RG_Rep277, RG_Contrl, RG_TA1, RG_repText
RGTextTemplat	Instream/	The name of	Example:	RG_Reptemp

Parameter Key	Area	Description	Example	Corresponding Java API Member
eFileRef	Response Gen	the template file to use if TEXT Document Type is specified in RGDocs. For details, see Controlfiles.	RGTextTemplateFileRef=SummReportTemplate.txt	
RGOptions	Instream/ Response Gen	String of Response Generator command-line arguments to use.	Example: RGOptions=-ge-y	RG_Options
RGTPAFileRef	Instream/ Response Gen	The name of the trading partner automation setup (.csv) file that is used to select a Response Generator Setup file. For details, see Controlfiles. Foresight REST API supports envelope-based Trading Partner Automation	Example: RGTPAFileRef=SampleTPA_DS_RG.csv	RG_ TradingPartner Automation

Parameter Key	Area	Description	Example	Corresponding Java API Member
		(TPA). The appropriate guideline and configuration files are automatically selected by using the IDs in the envelopes in the X12, EDIFACT, or TRADACOMS standard.		
DSProfile	Instream/ DocSplitting	Name of the DocSplitter INI file to use. For details, see Controlfiles.	Example: DSProfile=DSProfMedB.in	DS_Profile
DSTPAFile	Instream/ DocSplitting	Name of the trading partner automation setup (.csv) file that is used to select a DocSplitter INI file. For details, see Controlfiles. Foresight REST API supports envelope-based TPA. The	Example: DSTPAFile=SampleTPA_DS_RG.csv	DS_ TradingPartner Automation

Parameter Key	Area	Description	Example	Corresponding Java API Member
		appropriate guideline and configuration files are automatically selected by using the IDs in the envelopes in the X12, EDIFACT, or TRADACOMS standard.		
DSDebugFlag	Instream/ DocSplitting	Turns on debug messages in the DocSplitter report Y = Turn on debug messages N = No debug messages (default)		DS_ ReportDebugOpt, DS_ reportdebug
DSReportFormat	Instream/ DocSplitting	Type of the DocSplitter report format XML = XML report CSV = Text report (default)	Example: DSReportFormat=XML	DS_ ReportFormat

Parameter Key	Area	Description	Example	Corresponding Java API Member
DXProfile	Instream/DataExchange	Name of the Data Exchange setup file to use. For details, see Controlfiles.	Example: DXProfile=DXProfile1.ini	DX_Profile
TType (Required for operation involving translation: T, IT, TI. See Ops parameter)	Translation	Code that indicates the type of translation to be performed. Possible codes are: EE - EDI to EDI EX - EDI to XML EF - EDI to Flat XE - XML to EDI XF - XML to Flat FE - Flat to EDI FX - Flat to XML FF - Flat to Flat	TType=XE	Argument to convert routine
InDatafileRef	Translation	Name of the input (From) data file to use. For details, see Datafiles.	Example: InDatafileRef=ProdMedAData.edi	InputFile
MaxErrorsCount		When the validation errors reach the number, an	`MaxErrorsCount=100`	

Parameter Key	Area	Description	Example	Corresponding Java API Member
		error message displays in the Log, but the process proceeds. When the job is completed, the POST response as well as the message includes the same number of validation errors at most.		
copyResultsfile		When this parameter is included in the POST request, the results detail file (dtl) is copied to the shared file folder. Valid values: Y, or YES, and N or NO	`copyResultsfile=Y`	
publishReport	Transaction Insight	It enables Publish Information to Insight Reporting.	`publishReport=y`	

Parameter Key	Area	Description	Example	Corresponding Java API Member
		`publishReport` must be working with `CallbackTag` parameter. Valid values: y, or Y, and n, or N		
createTrk	Transaction Insight	It creates the track file. `createTrk` must work with `publishReport`. Valid values: y, or Y, and n, or N	`createTrk=y`	

JobID

Using the JobID parameter, the caller can specify the Job number to be used for that job. If omitted, the Foresight REST API creates one as usual, based on the Job Number algorithm type specified in the Foresight REST API config parameters. Currently, there is no limit in the format of the specified JobID. If a job with the specified ID exists on the Foresight REST API system, it appends the specified ID with a period followed by a counter until it finds a JobID that does not exist.

For example, if 'JobId=MYJOB' is specified and there already exists a `/jobs/MYJOB` resource, the REST API will try 'MYJOB.1', 'MYJOB.2', 'MYJOB.3', and so on.

Mode

The Foresight REST API normally runs in asynchronous mode. When a job request is

POSTed, Foresight REST API returns the assigned job and then process the job. The calling procedure checks back periodically to see if the job is done, and when the job is done, the user downloads the output files. A few more modes are available to give the calling routine more control over this process.

The 'Mode' parameter has the format.

```
‘Mode=n’ (Text)
```

or

```
'"Mode": "n"'
```

where ‘n’ is one of the following:

Mode	Description
ASYNC (Default)	Results in the standard asynchronous operation. When no mode is specified, this is the default mode.
ASYNCN	Similar to ASYNC, except that a notification is sent to an external server’s NotifyPostAddr Callback on job completion.
ASYNCO	Similar to ASYNCN, except that the notification is sent to an external server’s NotifyPostAddr Callback on job completion. It also includes a validation error summary (Instream), the contents of the response documents generated (RespGen), and the resulting files from any translation operations.
SYNC	Puts Foresight REST API into synchronous mode where it does return from the initial Job POST until the job is complete. On completion, the POST response contains the job completion information.
SYNCO	Similar to SYNC mode, except that the POST response also includes the contents of the following files (if any): Response files, Translated files.

Ops

Usually Foresight REST API can figure out the main operations to be done based on

the Job POST options specified. There may be cases, however, where this is impossible, for example passing just a DatafileRef and counting on the Callback to handle the guideline or map. When both operations are desired, the order of the two codes indicates the processing order.

For example: IT means Instream then Translate, TI means Translate then Instream. This parameter supports four values:

I	Instream operations (Validation, Response Generation, Doc Splitting, and Data Exchange)
T	Translation operations
IT	Instream operations are followed by translation operations. The Foresight REST API validates the input data file, and then translates error-free transaction sets. When Translator is not found, the "IT" operation fails, but the "I" operation runs.
TI	Translation operations are followed by Instream operations. The Foresight REST API translates the input data file, and then validates the resulting file. When Instream is not found, the "TI" operation fails, but the "T" operation runs.

Data

This parameter includes the data to be validated or translated as part of the Job POST body. When using the text parameter format, the data must be the last item in the Job POST, and must start with a single line ‘Data=’. The data then starts on the next line and runs to the end of the request. When using the JSON parameter format, the data parameter can occur anywhere in the Job POST body. When using the Data parameter, the DatafileRef parameter is optional. If present, the contents of the DatafileRef parameter are used in the Validation report as the data file name. If omitted, one is created by appending ‘_Data’ to the Job ID.

Examples:

Text Format

```
RGDocs=997,TA1 Mode=ASYNCN
Debug=y Data=
```

```
ISA*00*    *00*    *01*9012345720000 *01*9088877320000
*120111*1212*U*00401*000004321*0*T*:~
GS*HC*901234572000*908887732000*20120111*1615*1*X*004010X096A1~
ST*837*0001~
BHT*0019*00*3920394930203*20120111*1615*CH~
… SE*29*0001~
GE*1*1~ IEA*1*000004321~
```

JSON Format

```
{
“Ops”: “I”,
“Mode”: “SYNCO”,
“RGDocs”: “997,TA1”,

“Data”:
“ISA*00*    *00*    *01*9012345720000 *01*9088877320000

*120111*1212*U*00401*000004321*0*T*:~\nGS*HC*901234572000*908887732
000*20
120111*1615*1*X*004010X096A1~\nST*837*0001~\nBHT*0019*00*3920394930
203*20
120111*1615*CH~\n…\nSE*29*0001~\nGE*1*1~\nIEA*1*000004321~”
}
```

4. Possible Responses

Response	Description
200 OK	The job is started successfully. The response body contains the following information about the job: • Job ID • Date and time started • Guideline Name • Datafile Name The job ID is important because it is the key to further inquiries regarding the job and to the retrieval of any output files.

Response	Description
400 Bad Request	Problem with request (missing control file name, and so on); see the response body for reason.
409 Conflict	The control file cannot be overwritten; see the response body for additional information.
415 Unsupported Media Type	The controlfile PUT supports only a media type of ‘application/octet-stream’ and ‘text/plain’.
500 Internal Server Error	The server had problems writing a file; see response body for reason.

5. Examples
    a. http://<host-name>:<port>/ForesightREST/jobs

    Request Body:

    ```
    GuidelineRef=PDSA837I
    DatafileRef=837I_4010_H_6claims.txt
    RGDocs=997,277,824,TEXT
    RGTextTemplateFileRef=RGtemplate837I_c.txt
    RGOptions=-ge
    ```

    Response Body:

    ```
    Job #: J190528165545_00001
    Started at 05/28/2019 16:55:45.881
    GuidelineRef: PDSA837I
    DataFilename: 837I_4010_H_6claims.txt
    ```

    b. For JSON

    Request Body:

    ```
    {
    "GuidelineRef": "PCBS.std",
    "DatafileRef": "837I_4010_H_5provider.txt",
    "Mode": "SYNCO",
    ```

```
"Ops": "I",
"DSReportFormat": "xml",
"DSDebugFlag": "Y",
"DSProfile": "Content_Based_Split_Auto_Setup.ini"
}
```

Response Body:


```
{
    "invalidTxnCount": "1",
    "outputFiles": [
        "J231218224927_00001_DSInvalid_File0002.txt",
        "J231218224927_00001_DSReport.xml",
        "J231218224927_00001_DSValid_File0001.txt",
        "J231218224927_00001_DSValid_File0002.txt",
        "J231218224927_00001_DSValid_File0003.txt",
        "J231218224927_00001_DSValid_File0004.txt",
        "J231218224927_00001_DSValid_File0005.txt",
        "J231218224927_00001_VResult.txt",
        "Summary_J231218224927_00001_VResult.txt"
    ],
    "instreamMsgsBySeverity": "0,14,0,1,0,0,0",
    "started": "2023-12-18T22:49:27.819-06:00",
    "datafileRef": "837I_4010_H_5provider.txt",
    "completed": "2023-12-18T22:49:29.601-06:00",
    "mode": "SYNCO",
    "jobId": "J231218224927_00001",
    "ops": "I",
    "validTxnCount": "0",
    "instreamMsg": "",
    "instreamMsgsByWEDIType": "14,1,0,0,0,0,0,0,0",
    "instreamRCode": 100,
    "interchanges": [
        {
            "interchangeControlNumber": "000000545",
            "interchangeVersion": "00401",
            "groups": [
                {
                    "groupStartPos": "108",
                    "svalus": [],
                    "groupVersion": "004010X096A1",
                    "groupControlNumber": "545",
                    "groupEndPos": "7114",
                    "txns": [
```

```
                        {
                            "txnStartPos": "175",
                            "txnEndPos": "7103",
                            "svalus": [],
                            "txnControlNumber": "0545",
                            "evalus": [],
                            "errors": [
                                {
                                    "severity": "3",
                                    "segment": "DTP",
                                    "errorMessage": "Invalid
Date/Time '20110323-20110323': Should match pattern
'CCYYMMDD'",
                                    "segmentLocation": "101",
                                    "errorSegment":
"DTP*434*D8*20110323-20110323~",
                                    "loopGroup": "2300",
                                    "errorID": "31019",
                                    "loopGroupLocation": "1",
                                    "category": "Rejecting",
                                    "elementLocation": "3",
                                    "errorData": "20110323-
20110323",
                                    "element": "1251"
                                }
                            ],
                            "txnID": "837"
                        }
                    ]
                }
            ],
            "interchangeEndPos": "7133",
            "interchangeStartPos": "0",
            "txns": []
        }
    ],
    "status": "COMPLETE"
}
```


GET

The GET operation is used to.

- Get information about a specific job on the Foresight REST API server, or
- Get a list of all of the jobs on the Foresight REST API server.

For specific job GETs, the following information is returned in the response body:

- Job ID
- Job Status: Queued, Submitted, In Progress, Completed
- Date and time started
- Date and time completed
- Guideline used
- Data file used
- Validation return code and error message, if one
- Count of messages by severity
- Count of messages by type
- List of output files

1. REST Path
    a. For a specific job

    ```
    .../jobs/jobId
    ```

    b. For a list of all jobs

    ```
    .../jobs
    ```

3. Optional Parameters

    archive=Y

    Used when requesting a list of all jobs to include those jobs that have completed and have been moved off the ‘Job Board’ onto disk. If requesting information on a specific job, this parameter is ignored.

4. Request Body: (Empty)
5. Possible Responses

Response	Description
200 OK	Job information retrieval is successful; the response body contains information.

Response	Description
400 Bad Request	Problem with the request (missing job ID, and so on); see the response body for reason.
404 Not Found	Specified job cannot be found on Server
415 Unsupported Media Type	Control file GET supports only media types of 'text/plain'.
500 Internal Server Error	Server had problems reading information; see the response body for reason.

5. Examples
   a. http://<host-name>:<port>/ForesightREST/jobs/J190528165545_00001

      Response Body:

```
JobID:J190528165545_00001
Status:COMPLETE
Started:2019-05-28 16:55:45.881-0400
Completed:2019-05-28 16:56:03.269-0400 Guide:PDSA837I
Data:837I_4010_H_6claims.txt Ops:32
InstreamRCode:100 InstreamMsg:
MsgsBySeverity:0,23,0,4,0,0,0
MsgsByWEDIType:23,4,0,0,0,0,0,0,0,0
--Output Files-- J190528165545_00001_RG277.txt J190528165545_
00001_RG824.txt J190528165545_00001_RG997.txt J190528165545_
00001_RGText.txt
J190528165545_00001_VResult.txt Summary_J190528165545_00001_
VResult.txt
```

   b. http://SERVX:8080/ForesightREST/jobs?archive=y

      Response Body:

```
...JOB BOARD...
J190528165545_00003    COMPLETE    started:190528165545
completed:190528165545 J190528165345_00002    PROCESSING
started:190528165545
J190528165225_00001    COMPLETE    started:190528165545
completed:190528165545
```

```
...ARCHIVE... J190507164911_00011 J190522162506_00001
J190522165258_00002 J190522170723_00003
```

DELETE

The DELETE operation is used to remove a job resource from the REST server, along with all of its Output files.

1. REST Path: `.../jobs/jobId`
2. Request Body: (Empty)
3. Possible Responses

Response	Description
200 OK	The specified job is deleted successfully.
400 Bad Request	Problem with request (missing job ID, and so on); see response body for reason.
409 Conflict	The specified job resource cannot be deleted; see the response body for additional information.
500 Internal Server Error	The server had problems deleting the job; see response body for reason.

Job Output Files

The purpose of a job is to produce output files. Job Output File resources are created during job processing initiated through a POST process.

Two REST API operations are supported for Job Output File resources: GET and DELETE

GET

The GET operation is used to:

- Get a specific output file from a particular job on the REST API server
- Get a list of all the output files generated by a particular job on the server

For specific Output File GETs, the contents of the requested file will be returned in the response body (media type=Application/Octet-Stream).

When a list of all output files is requested, the following information is provided for each output file found for the specified job:

- File name
- File Size
- File Creation Date and Time
- File Last Modification Date and Time

Note: Fields are tab delimited.

1. REST Path
    a. For a specific output file for job ‘jobId’
    `.../jobs/jobId/output/outputFileName`
    b. For a list of all output files for a specified job ‘jobId’
    `.../jobs/jobId/output`
2. Optional Parameters
    a. (None)
3. Request Body: (Empty)
4. Possible Responses

Response	Description
200 OK	Output file information retrieval is successful; the response body contains information.
400 Bad Request	Problem with request (missing job ID, output file name, and so on); see the response body for reason.

Response	Description
404 Not Found	The specified job/output file cannot be found on the server.
415 Unsupported Media Type	Output file GET supports only media types of 'Application/Octet-Stream'.
500 Internal Server Error	The server had problems reading information; see response body for reason.

5. Examples

1. http://<host-name>:</ForesightREST/jobs/J190528165545_00001/output

   Response Body:

```
J190528165545_00001_RG277.txt 889 2019-05-28T20:56:03.164831Z 2019-
05-28T20:56:03.304824Z J190528165545_00001_RG824.txt 0 2019-05-
28T20:56:03.165871Z 2019-05-28T20:56:03.165871Z J190528165545_
00001_RG997.txt 681 2019-05-28T20:56:03.161896Z 2019-05-
28T20:56:03.229838Z J190528165545_00001_RGText.txt 1365 2019-05-
28T20:56:03.167827Z 2019-05-28T20:56:03.305916Z J190528165545_
00001_VResult.txt 16015 2019-05-28T20:56:02.603854Z 2019-05-
28T20:56:02.893816Z Summary_J190528165545_00001_VResult.txt 466
2019-05-28T20:56:02.605864Z 2019-05-28T20:56:02.895826Z
```

2. http://SERVX:8080/ForesightREST/jobs/J190528165545_00001/output/J190528165545_00001_RG997.txt

   Response Body:

```
ISA*00* *00* *01*HORIZONE *01*HILLSDALEHOSP
*190528*1656*U*00401*000000001*0*P*:~
GS*FA*HORIZONE*HILLSDALEHOSP*20190528*165603*1*X*004010~
ST*997*0001~
AK1*HC*1~ AK2*837*0001~ AK3*DTP*58*2300*8~ AK4*3*1251*8*20120212~
AK3*DTP*77*2300*8~
AK4*3*1251*6~ AK4*3*1251*8*12~ AK3*DTP*98*2300*8~
AK4*3*1251*8*20120212~
AK5*R*5~ AK9*R*1*1*0~ SE*13*0001~
GE*1*1~ IEA*1*000000001~
ISA*00* *00* *01*9088877320000 *01*9012345720000
*190528*1656*U*00401*000000002*0*P*:~
```

```
GS*FA*908887732000*901234572000*20190528*165603*2*X*004010~
ST*997*0002~
AK1*HC*2~ AK2*837*0002~
AK5*A~ AK9*A*1*1*1~ SE*6*0002~ GE*1*2~ IEA*1*000000002~
```

DELETE

The DELETE operation is used to remove a job Output File resource from the Foresight REST API server.

1. REST Path: `.../jobs/jobId/output/outputFilename`
2. Request Body: (Empty)
3. Possible Responses

Response	Description
200 OK	The specified output file for the job is deleted successfully.
400 Bad Request	Problem with request (missing job ID, output file name, and so on); see the response body for reason.
409 Conflict	The specified output file resource cannot be deleted; see the response body for additional information.
500 Internal Server Error	The server had problems deleting the output file; see the response body for reason.

Callbacks

A Callback is a request that originates from the Foresight REST API server and is sent to the client in response to an earlier request sent by that client. Foresight REST API supports 4 main types of callbacks.

Mid-Processing Callback

You can initiate a callback with Foresight REST API when starting a Validation or Translation process. This callback occurs after the data file header has been parsed, and it sends information about the data to another process that can then return one or more guidelines, map, and so on, for use in the subsequent Validation and Translation processing.

Data sent to external system via POST from Foresight REST API Instream/Translator Callback

JSON Format


```
{
"sourceOp": "T",
"jobId": "J200716171357_00001",
"tag": "Inbound",
"protocol": "X12", "txnInfo": {
"interchange": { "date": "110924",
"receiverQualifier": "01",
"senderID": "9012345720000",
"receiverID": "9088877320000",
"senderQualifier": "01",
"ctlNo": "000000370",
"date": "110924",
"time": "0947",
"version": "00501"
},
"functionalGroup": { "date": "20110924",
"receiver": "908887732000",
"sender": "901234572000",
```

```
"ctlNo": "370",
"groupID": "HC",
"time": "1615", "version": "005010X223A2"
},
"transactionSet": { "transactionSetID": "837"
}
},
"textInfo": {
"receiverDomain": "01",
"senderID": "123456789",
"receiverID": "987654321", "messageType": "PEDIDOAUTEXT",
"recordDelimiter": "0x27", "fieldDelimiter": "", "messageGroup": "TEXT",
"key": "S1",
"senderDomain": "01"

}
}
```


Text Format


```
<?xml version="1.0" encoding="UTF-8"?
<Info id="X12">
<Interchange>
<SenderQualifier>01</SenderQualifier>
<Sender>9012345720000</Sender>
<ReceiverQualifier>01</ReceiverQualifier>
<Receiver>9088877320000</Receiver>
<Date>110924</Date>
<Time>0947</Time>
<CtlNo>000000370</CtlNo>
<ICVersion>00501</ICVersion>
</Interchange>
<FunctionalGroup>
<ID>HC</ID>
<Sender>901234572000</Sender>
<Receiver>908887732000</Receiver>
<Date>20110924</Date>
<Time>1615</Time>
<CtlNo>370</CtlNo>
<Version>005010X223A2</Version>
</FunctionalGroup>
<TransactionSet>
<TransactionSetID>837</TransactionSetID>
</TransactionSet>
</Info>
<Source>
<Op>T</Op>
```

```
<Job>J200716103132_00002</Job>
<Tag>Inbound</Tag>
</Source>
```

Data to be returned to Foresight REST API

The POST response from the external system supports returning data to Foresight REST API Instream/Translator Callback. The following examples show the JSON and text format of the data.

JSON Format

```
{
"senderName": "string", "receiverName": "string", "operationInfo": {
"validationGuidelineID": "string", "validationProfileID": "string",
"translationMapID": "string", "sourceGuidelineID": "string",
"targetGuidelineID": "string", "translationType": "string"
}
}
```

Text Format

```
senderName=string receiverName=string validationGuidelineID=string
validationProfileID=string translationMapID=string
sourceGuidelineID=string targetGuidelineID=string translationType=code
```

Parameters	Description
`senderName`	Optional Name of the person who sends the data.
`receiverName`	Optional Name of the person who receives the data.
`validationGuidelineID`	Required This is required when validation is to be performed. If both translation and validation are to be performed and this parameter is omitted, the target guideline is used for validation.

Parameters	Description
`validationProfileID`	Optional ProfileID is required to perform validation.
`translationMapID`	Required Map ID, is required to perform translation.
`sourceGuidelineID`	Optional If omitted, the sourceGuideline from the map specified in mapID is used.
`targetGuidelineID`	Optional If omitted, a target guideline from the map specified in mapID is used.
`translationType`	Optional If omitted, it is automatically determined from the source and target guidelines. The valid codes include: EE - EDI to EDI EX - EDI to XML EF - EDI to Flat XE - XML to EDI XF - XML to Flat FE - Flat to EDI FX - Flat to XML FF - Flat to Flat

File Request Callback

When Foresight REST API cannot find a specified file on its system, it sends a File Request GET to an external process on job completion using the File Request Callback. The file name and resource type are passed in the query:

http://xxx/fileName?type=t

Where:

xxx	File Request Callback server URL.
filename	The name of the file requested.
t	A code corresponding to the resource type of the file: c = controlfile d = datafile g = guideline m = map

For example:

GET http://<host-name>:<port>/CallbackTesterREST/FileReq/my4010_837.std?type=g is requesting the guideline file my4010_837.std.

Notification Callback

Foresight REST API can send a notification to an external process on job completion using the NotifyCallback. The information contained in the Notification POST includes information about the job, various status codes, and the output files it produced.

Data sent to external system via POST from Foresight REST API Notification Callback

JSON Format

```
{
"sourceOp": "string", // I or T "jobId": "string",     // Job ID

"status": "string",    // COMPLETE (only notification event so far)
"started": "string", // yyyy-mm-dd hh:mm:ss.mmm+/-tttt "completed":
"string",       // yyyy-mm-dd hh:mm:ss.mmm+/-tttt
"ops": "string",       // I, T, IT, TI "instreamRCode": "string",
"instreamMsg": "string",
"instreamMsgsBySeverity": "string", // n,n,n,n,n,n,n
"instreamMsgsByWEDIType": "string", // n,n,n,n,n,n,n
```

```
"translatorRCode": "string", "translatorMsg": "string",

"outputFiles": // Used only when output files are present [
"fileName1", "fileName2", "fileName3",
...
"fileNameN",
]
}
```

TEXT Format

```
SourceOp:string // I or T JobID:string  // Job ID

Status:string  // COMPLETE (only notification event so far)
Started:string  // yyyy-mm-dd hh:mm:ss.mmm+/-tttt Completed:string      //
yyyy-mm-dd hh:mm:ss.mmm+/-tttt

Ops:string     // I, T, IT, TI

InstreamRCode:string InstreamMsg:string
InstreamMsgsBySeverity:string  // n,n,n,n,n,n,n
InstreamMsgsByWEDIType:string   // n,n,n,n,n,n,n

TranslatorRCode:string TranslatorMsg:string

--OutputFiles-- fileName1 fileName2 fileName3
...
filename
```

Data to be returned to Foresight REST API via POST response from External System to Foresight REST API Notification Callback

No data is returned. Instead, an HTTP return code indicates success or failure.

Message Callback

Foresight REST API can POST messages to an external process at various times using Message Callback. The information contained in the Message POST includes the Job ID, Business Area, Message Title and Description, the message date and time, information about the job, various status codes, and the output files it produced. The only format that Foresight REST API currently handles is the BCCE/AuditSafe format.

Note: The maximum number of audit_event `INSTREAM_VALIDATION_ERROR` is 20.

Data sent to external system via POST from Foresight REST API Message Callback

```
[{
"transaction_id":"J230707142635_00001", // Job ID "biz_proc":"FR_J",   //
Business Area
"audit_event":"Job Completed", // Title "event_desc":"Job posting
process completed", // Description "event_timestamp":"2023-07-
07T14:26:36.440-04:00", // Date and time "event_status":"FAILED"        //
Status
}]
```

Data to be returned to Foresight REST API via POST response from External System to Foresight REST API Message Callback

No data is returned. Instead, an HTTP return code indicates success or failure.

Foresight REST API Configuration

To check the Foresight REST API configuration user can run the Get request with status=y parameter that is.

```
GET http://localhost:8080/ForesightREST?status=Y
```

The user gets all the config parameters with values in response. The response is as follows:

```
Foresight REST API Version 1.2.0 build v02
Status at 2024-09-04 11:28:16 UTC
Started: 2024-08-23 06:18:12 UTC
Jobs Run since start: 0
Instream Thread Pool Count: 5 (Max: 128)
Translator Thread Pool Count: 9 (Max: 128)
Max Validation Errors Count: 100 (Max: 1720000)
Job ID Type: Timestamp (yymmddhhmmss_nnnnn)
Guideline ID Match Required on PUT: true
Logging Level: 1
Job Logging: false
Job Retention: Forever
Skipped Transaction sets for PublishReport: 837,835
FSRest.ini Directory: /usr/local/tomcat/TibcoFSRest/
Instream Library Status: Loaded
Instream Library Version: 9.2.0 [Build 012r(64 bit): 11/19/2021]
Instream Edition:   Standard
Code Table: not found.
Guideline Version: ValidatorVersion=9.2.0.0
Instream Base Path: /usr/share/tomcat/tibco/instream/fseng/
Instream Ini Path: /usr/share/tomcat/tibco/instream/fseng/bin/
Translator Library Status: Loaded
Translator Library Version: 4.1.0 Build 007 [Build 007r(64 bit):
10/19/2020]
Translator Base Path: /usr/share/tomcat/tibco/instream/fseng/
Translator Ini Path: /usr/share/tomcat/tibco/instream/fseng/bin/
Guidelines Path: /usr/local/tomcat/TibcoFSRest/guidelines/
Maps Path: /usr/local/tomcat/TibcoFSRest/maps/
DataFiles Path: /usr/local/tomcat/TibcoFSRest/datafiles/
ControlFiles Path: /usr/local/tomcat/TibcoFSRest/controlfiles/
Jobs Base Path: /var/cache/tomcat/temp/TibcoFSRestJobs/
External System Links:
Validation Callback POST: Address=(none) Format=Unknown Format 0
```

```
Translation Callback POST: Address=(none) Format=Unknown Format 0
Default Buffer=2048
Notification POST: Address=(none) Format=Unknown Format 0
Message POST: Address=(none) Format=Default
File Request Callback POST: Address=(none) Format=(none)
Current Working Directory: /usr/local/tomcat
Operating System: Linux
Properties:
  APIKeyExpireTime=604800 (Src:ini)
  DefaultResponseMediaType=application/json (Src:ini)
  DefaultThreadLocking=false (Src:default)
  FileReqCallbackFormat=TEXT (Src:default)
  FileRequestCaching=ALL (Src:ini)
  FileRequestCallbackAddr= (Src:default)
  FileRequestCallbackSecurity= (Src:default)
  GuidelineIDMatchRequired=Y (Src:ini)
  InstreamIniPath=/usr/share/tomcat/tibco/instream/fseng/bin/ (Src:env
[FSINSTREAMINI])
  InstreamMainPath=/usr/share/tomcat/tibco/instream/fseng/ (Src:calc)
  InstreamThreadCount=5 (Src:ini)
  JobIdType=S (Src:ini)
  JobRetention=ALL (Src:ini)
  LogLevel=1 (Src:ini)
  MaxErrorsCount=100 (Src:ini)
  MessageFormat=AuditSafe (Src:ini)
  MessagePostAddr= (Src:default)
  MessagePostFormat=JSON (Src:ini)
  MessagePostSecurity= (Src:default)
  NotifyPostAddr= (Src:default)
  NotifyPostFormat=JSON (Src:ini)
  NotifyPostSecurity= (Src:default)
  OutputJobMessagesSection=Y (Src:ini)
  RestControlPath=/usr/local/tomcat/TibcoFSRest/controlfiles/ (Src:ini)
  RestDataPath=/usr/local/tomcat/TibcoFSRest/datafiles/ (Src:ini)
  RestGuidelinesPath=/usr/local/tomcat/TibcoFSRest/guidelines/ (Src:ini)
  RestJobsPath=/var/cache/tomcat/temp/TibcoFSRestJobs/ (Src:ini)
  RestMapsPath=/usr/local/tomcat/TibcoFSRest/maps/ (Src:ini)
  RestRetainJobsPath=/usr/local/tomcat/TibcoFSRest/retainjobs/ (Src:ini)
  SkipTS=837,835 (Src:ini)
  TIInboundPath=/var/cache/tomcat/temp/inbound (Src:ini)
  TIOutboundPath=/var/cache/tomcat/temp/outbound (Src:ini)
  TimestampPattern=yyyy-MM-dd'T'HH:mm:ss.SSSXXX (Src:ini)
  TransCallbackAddr= (Src:default)
  TransCallbackFormat=JSON (Src:ini)
  TransCallbackSecurity= (Src:default)
  TranslatorIniPath=/usr/share/tomcat/tibco/instream/fseng/bin/ (Src:env
[FSTRANSLATORINI])
```

```
TranslatorMainPath=/usr/share/tomcat/tibco/instream/fseng/ (Src:calc)
TranslatorThreadCount=9 (Src:ini)
ValCallbackAddr= (Src:default)
ValCallbackFormat=JSON (Src:ini)
ValCallbackSecurity= (Src:default)
jdbc.driver=oracle.jdbc.driver.OracleDriver (Src:ini)
jdbc.platform=oracle (Src:ini)
jdbc.url=jdbc:oracle:thin:@<Database Host>:1521/<SID> (Src:ini)
jdbc.username=<Database Username> (Src:ini)
secureWithAPIKey=y (Src:ini)
```

The status response parameters are described below. Note that the response values of config properties can be changed using the corresponding environment variable name. Unless otherwise specified, the overriding environment variable name is the parameter keyword converted to all caps with the prefix `FSREST_` added. For example, the InstreamThreadCount parameter setting is overridden by the contents of the FSREST_INSTREAMTHREADCOUNT environment variable.

Keyword	Response Description
`InstreamThreadCount`	The number of processing threads in the Instream thread pool. It controls the number of Instream jobs that can be processed simultaneously. Set higher values for heavy loads, but too high can bog down the machine. The maximum setting is 128. Example: InstreamThreadCount=5 The default value is 3.
`TranslatorThreadCount`	The number of processing threads in the Translator thread pool. It controls the number of translation jobs that can be processed simultaneously. You can set higher values for heavy loads, but too high can bog down the machine. The maximum setting is 128. Example: TranslatorThreadCount=9 The default value is 3.
`TimestampPattern`	Sets the pattern for date and time output in the resource GET information.

Keyword	Response Description
	Note: AuditSafe requires the timestamp format 'nnnn-nn-nnTnn:nn:nn.nnn+/-zzz' Example: TimestampPattern=yyyy-MM-dd HH:mm:ss.SSSZ Default: yyyy-MM-dd'T'HH:mm:ss.SSSXXX
`InstreamMainPath`	Directory path to the top of the Instream installation. Example: InstreamMainPath=c:/tibco/instream/9.0 Environment Variable Override: FSINSTREAMMAINDIR
`InstreamIniPath`	Directory where the main Instream dir.ini file is located. Example: InstreamIniPath=c:/tibco/instream/9.0/bin Environment Variable Override: FSINSTREAMINI
`TranslatorMainPath`	Directory to the top of the Translator installation. Example: TranslatorMainPath=c:/tibco/translator/4.0 Environment Variable Override: FSTRANSLATORMAINDIR
`TranslatorIniPath`	Directory where the main Translator dir.ini file is located. Example: TranslatorIniPath=c:/tibco/translator/4.0/bin Environment Variable Override: FSTRANSLATORINI
`JobIdType`	Controls the type of Job ID required. ‘S’ uses the current date and time along with a counter to create a Job ID of the format Jyymmddhhmmss_nnnnn. Example: J190507164911_00001 ‘U’ uses a full UUID for the Job ID. Example: Jbca8888d-b27a-4cb5-9aeb-b4a1655bd066 Using ‘U’ is preferred when running more than one instance of

Keyword	Response Description
	the REST API on a single machine as the chance of generating identical IDs is greatly reduced. This allows the sharing of folder structures between the instances. Example: JobIdType=S Default: S
RestControlPath	Directory where REST 'controlfiles' resources are to be stored. Read/Write permissions are required. Environment Variable Override: FSREST_CONTROLDIR
RestDataPath	Directory where REST 'datafiles' resources are to be stored. Read/Write permissions are required. Example: RestDataPath=C:\FSRest\datafiles Environment Variable Override: FSREST_DATADIR
RestGuidelinesPath	Directory where REST 'guidelines' resources are to be stored. Read/Write permissions are required. Environment Variable Override: FSREST_GUIDELINEDIR
RestJobsPath	Directory where REST 'jobs' resources are to be stored. Read/Write and Create Directory permissions are required. Example: RestJobsPath=C:\FSRest\jobs Environment Variable Override: FSREST_JOBSDIR
RestMapsPath	Directory where REST 'maps' resources are to be stored. Read/Write permissions are required. Environment Variable Override: FSREST_MAPDIR
OutputJobMessagesSection	Include Messages section in Job SYNC responses and Notifications
GuidelineIDMatchRequire	Indicates if any uploaded guideline must have an internal ID

Keyword	Response Description
d	that matches its external filename. Default: y
ValCallbackAddr	The HTTP address of the server that processes the Instream Callback HTTP POST request. Example: ValCallbackAddr=http://<host-name>:<port>/CallbackTesterREST/ValCB
ValCallbackFormat	Indicates the content format for POSTs to, and replies from, the Instream Callback process specified in ValCallbackAddr. Valid Values J or JSON = application/json T or TEXT = text/plain Default: TEXT
ValCallbackSecurity	Specifies the type of security with any parameters in use between Foresight REST API and the server/application processing the Instream Callback POSTs. The general format of this parameter is: Type: Info Where type is the code for the type of security implemented, and Info is the additional data required for that type of security. Currently, the following types of security are supported: • BASIC: Basic security, where the information contains the user ID and password separated by a colon (:). • BASIC64: Basic security with encode user ID and password, where the information contains the Base 64 encoded user ID and password. Examples: • ValCallbackSecurity=BASIC:admin3:myPassword

Keyword	Response Description
	• ValCallbackSecurity=BASIC64:YWRtaW23aXdheQ==
TransCallbackAddr	The HTTP address of the server that processes the Translator Callback HTTP POST request. Example: TransCallbackAddr=http://<host-name>:<port>/LocalApp/HandleFSRest
TransCallbackFormat	Indicates the content format for POSTs to, and replies from, the Translator Callback process specified in TransCallbackAddr. Valid Values J or JSON = application/json T or TEXT = text/plain Default: TEXT
TransCallbackSecurity	Specifies the type of security with any parameters in use between the Foresight REST API and the server/application processing the Translator Callback POSTs. (See ValCallbackSecurity description for details.)
NotifyPostAddr	The HTTP address of the server that processes the Notification Callback HTTP POST request. Example: NotifyPostAddr=http://<host-name>:<port>/myApp/CatchFSRestNotify
NotifyPostFormat	Indicates the content format for POSTs to, and replies from, the Notification Callback process specified in NotifyPostAddr. Valid Values J or JSON = application/json T or TEXT = text/plain Default: TEXT
NotifyPostSecurity	Specifies the type of security with any parameters in use between REST API and the server/application processing the notification Callback POSTs. (See ValCallbackSecurity

Keyword	Response Description
	description for details.)
MessagePostAddr	The HTTP address of the server that processes the Message Callback HTTP POST request. MessagePostAddr=http://<host-name>:<port>/myLogger/catchMessage
MessagePostFormat	Indicates the content format for POSTs to, and replies from, the Message Callback process specified in MessagePostAddr. Valid Values J or JSON = application/json T or TEXT = text/plain Default: TEXT
MessagePostSecurity	Specifies the type of security with any parameters in use between Foresight REST API and the server/application processing the message Callback POSTs. (See ValCallbackSecurity description for details.)
MessageFormat	Controls the format of the external messages that are sent to the messaging POST address. There are two values for MessageFormat: **default** value passes the message as the body of the message POST, and **AuditSafe** formats the message for AuditSafe, including passing the various AuditSafe fields. **Note:** If this parameter is not present, or has values other than the two defined values, then it takes the default value. These values are not case-sensitive.
FileRequestCallbackAddr	The HTTP address of the server that processes the Message Callback HTTP POST request. Example:

Keyword	Response Description
	FileRequestCallbackAddr=http://<host-name>:<port>/myLogger/catchMessage
`FileReqCallbackFormat`	Indicates the content format for POSTs to, and replies from, the File Request Callback process specified in FileRequestCallbackAddr. Valid Values J or JSON = application/json T or TEXT = text/plain Default: TEXT
`FileRequestCallbackSecurity`	Specifies the type of security with any parameters in use between Foresight REST API and the server/application processing the File Request Callback POSTs. (See ValCallbackSecurity description for details.)
`FileRequestCaching`	Cache files are returned to Foresight REST API via the FileRequest Callback. Files are cached by writing them to their respective Foresight REST API resource directory. For example, Guidelines are cached in the Foresight REST API Guideline directory, Maps are cached in the Foresight REST API Maps directory. The caching is controlled using the FileRequestCaching parameter, which can be set in the `FSREST.ini` file. This parameter can be 'All' or a string of letters identifying the types of files to be cached. Example: A FileRequestCaching parameter of 'GMC' caches guidelines, maps, and control files. G = guidelines, M = maps, C = control files, D = data files. The order and casing of the letter are not significant, that is, GMC, mcg, and cgM mean the same thing. Specifying 'All' or 'GMCD' are the same in the FileRequestCaching parameter. If the FileRequestCaching parameter is set to 'None', then no

Keyword	Response Description
	files received via the File Request Callback are cached. As with the other parameters, the FileRequestCaching parameter can be overridden with the contents of the environment variable FSREST_FILEREQUESTCACHING. **Note:** The FileRequestCaching setting has no effect if the FileRequestCallback parameter is not used.
`LogLevel`	Controls the verbosity level of log events: 0 = Errors 1 = Warnings 2 = Recommendations 3 = Informational 5 = Debug Default: 1
`DefaultThreadLocking`	Sets the default setting for the jobs.
`JobRetention`	Controls whether job files, including output files, are kept on the system after a SYNCO or ASYNCO job is completed and their output is returned or posted. Values: All: All jobs and their output are kept on the system (default) NoSYNCO: All jobs run with mode SYNCO has their files deleted once the job POST has been returned to the submitter. NoASYNCO: All jobs run with mode ASYNCO has their files deleted once the job results and files have been POSTed and acknowledged by the NotifyCallback receiver. **Note:** Both NoSYNCO and NoASYNCO can be specified, separated by a comma.

Keyword	Response Description
	Example: JobRetention=NoSYNCO
`secureWithAPIKey`	Secures your application using an API key for authentication.
`APIKeyExpireTime`	Expire time (in seconds) for API key.
`jdbc.platform`	MYSQL or Oracle If you are using an Oracle database, add its connector lib path to the `database.driver.path`
`jdbc.driver`	Database driver URL
`jdbc.url`	Database driver URL
`jdbc.username`	Database username
`jdbc.databasename`	Database name for MYSQL or Oracle database. **Note:** Property is specific to the Mysql database.
`RestRetainJobsPath`	Directory where `RetainJobs` resources are to be stored. Read/Write permissions are required. The parameter can be update using `updateEnv` request. Following is the default `RetainJobs` path: `RestRetainJobsPath=/usr/local/tomcat/TibcoFSRest/retainjobs`
`TIInboundPath`	Inbound path of the Foresight Transaction Insight. You can update this using `updateEnv` request.
`TIOutboundPath`	Outbound path of the Foresight Transaction Insight. You can update this using `updateEnv` request.
`skipTS`	It skips one or multiple transaction set types for the copy

Keyword	Response Description
	operation. The default value is none, and could be set with the transaction set names, separated by commas. You can update this using `updateEnv` request.

Other Environment Variables

FSTRANSLATORINI	Contains path to the Translator Installation's bin directory.
InstreamIniPath	`FSINSTREAMINI`
InstreamMainPath	`FSINSTREAMMAINDIR`
TranslatorIniPath	`FSTRANSLATORINI`
TranslatorMainPath	`FSTRANSLATORMAINDIR`

Changing Foresight REST API Config Parameters

You can now change certain parameters without restarting the Foresight REST API server. This is done via a PUT operation to the main REST API address with the 'updateEnv=y' parameter. The body of PUT contains the parameters to change. You can change the following parameters:

`TimestampPattern`	`TransCallbackFormat`	`FileReqCallbackFormat`
`JobIdType`	`NotifyCallbackAddr`	`FileRequestCaching`
`GuidelineIDMatchRequired`	`NotifyPostFormat`	`LogLevel`
`ValCallbackAddr`	`MessageCallbackAddr`	`JobRetention`
`ValCallbackFormat`	`MessagePostFormat`	`DefaultResponseMediaType`
`TransCallbackAddr`	`FileReqCallbackAddr`	`NotifyPostAddr`
`MessagePostAddr`	`RestRetainJobsPath`	

For example:

http:.../ForesightREST?updateEnv=y

Body

`LogLevel`=0

`ValCallbackAddr`=http://NA-DUB-DOCKER1:8080/BCStubREST/ValCB

`NotifyPostAddr`=http://NA-DUB-DOCKER1:8080/BCStubREST/Notify

`MessagePostAddr`=http://NA-DUB-DOCKER1:8080/BCStubREST/Message

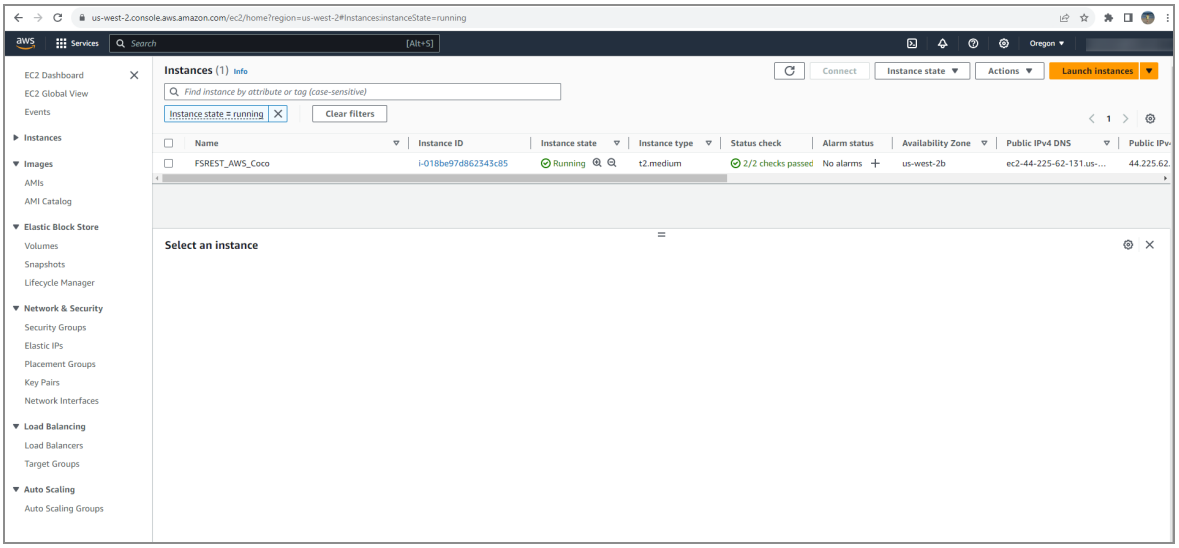
Note: Tags are case-insensitive and blank lines are ignored. Invalid entries are noted in the PUT response and no changes are made.

AWS Buckets for TIBCO Foresight REST API Data Storage

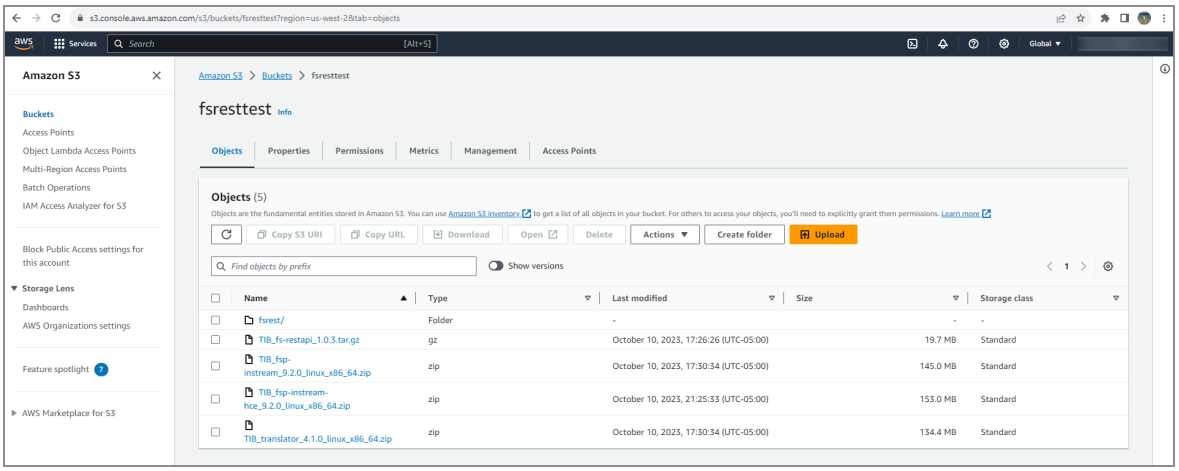
In this release, you can now use AWS buckets for storing Foresight REST API data. To store data in AWS buckets, follow the given steps:

Procedure

1. Sign in to AWS. Create an EC2 instance of the supported OS, for example: Ubuntu OS.



2. Download AWS CLI and connect to AWS.
3. Install Docker CE on the Ubuntu machine. For details, see Install Docker Engine on Ubuntu.
4. Create S3 bucket, upload Foresight Instream, Foresight Translator, Foresight REST API build.

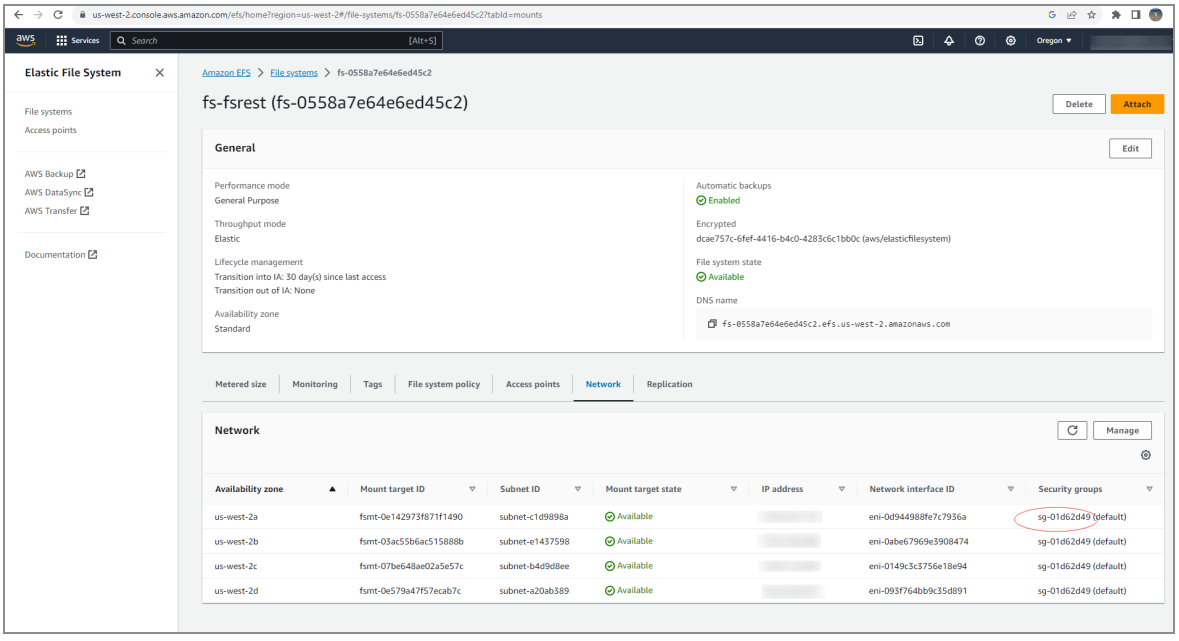


5. Copy all the builds to EC2 instance.

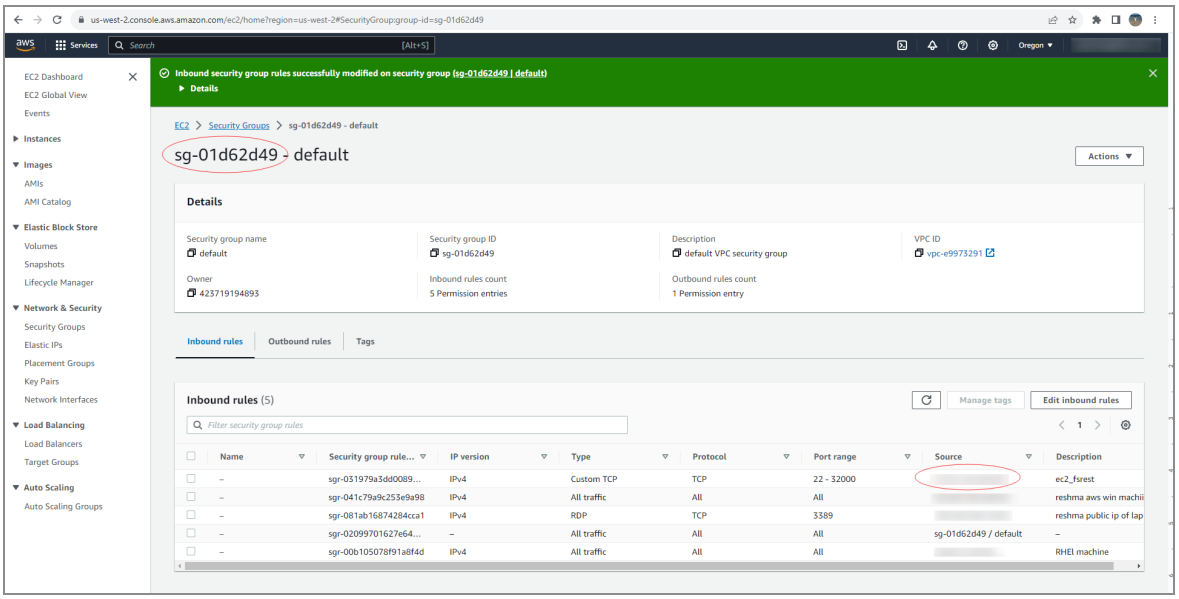
```
aws s3 cp s3://fsresttest/TIB_fsp-instream-hce_9.2.0_linux_x86_
64.zip
aws s3 cp s3://fsresttest/TIB_translator_4.1.0_linux_x86_64.zip
aws s3 cp s3://fsresttest/TIB_translator_4.1.0_linux_x86_64.zip
```

6. Setup Foresight REST API 1.2.0 on the Ubuntu machine.

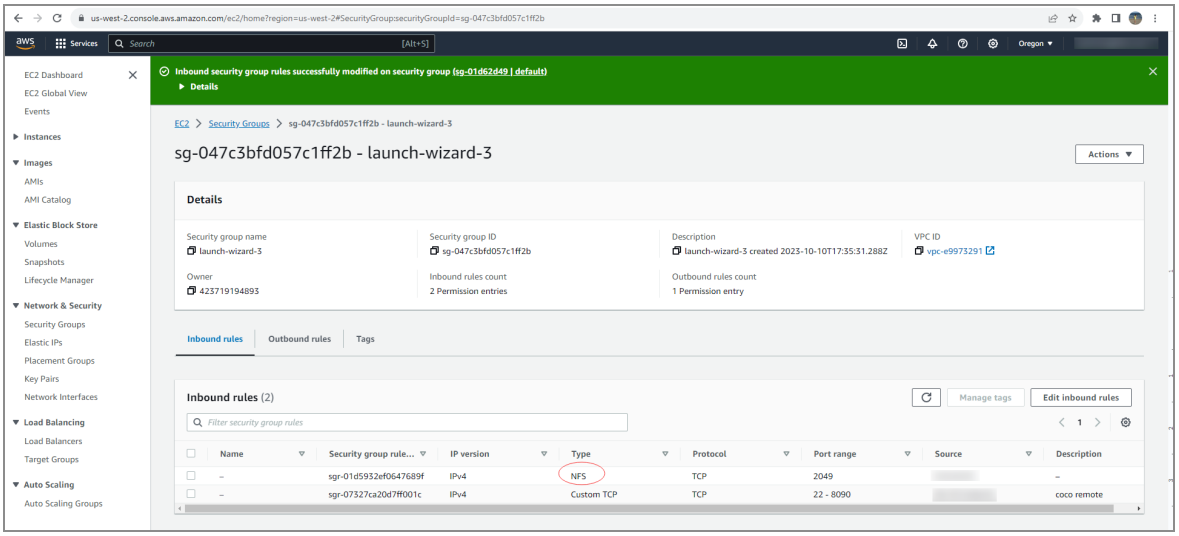
7. Create an Elastic File System (EFS) in the same region.



Add the EC2 instance private IP into the security group of the EFS.



8. In Inbound rules of the EC2 instance security group, add NFS in type.



9. Mount EFS on EC2 instance.

```
cd ~
sudo mkdir efs
cd efs
sudo mkdir controlfiles
sudo mkdir datafiles
sudo mkdir guidelines
sudo mkdir maps
```

```
cd ~
sudo mount -t nfs4 -o
nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,no
resvport $efs_dns:/ efs
```

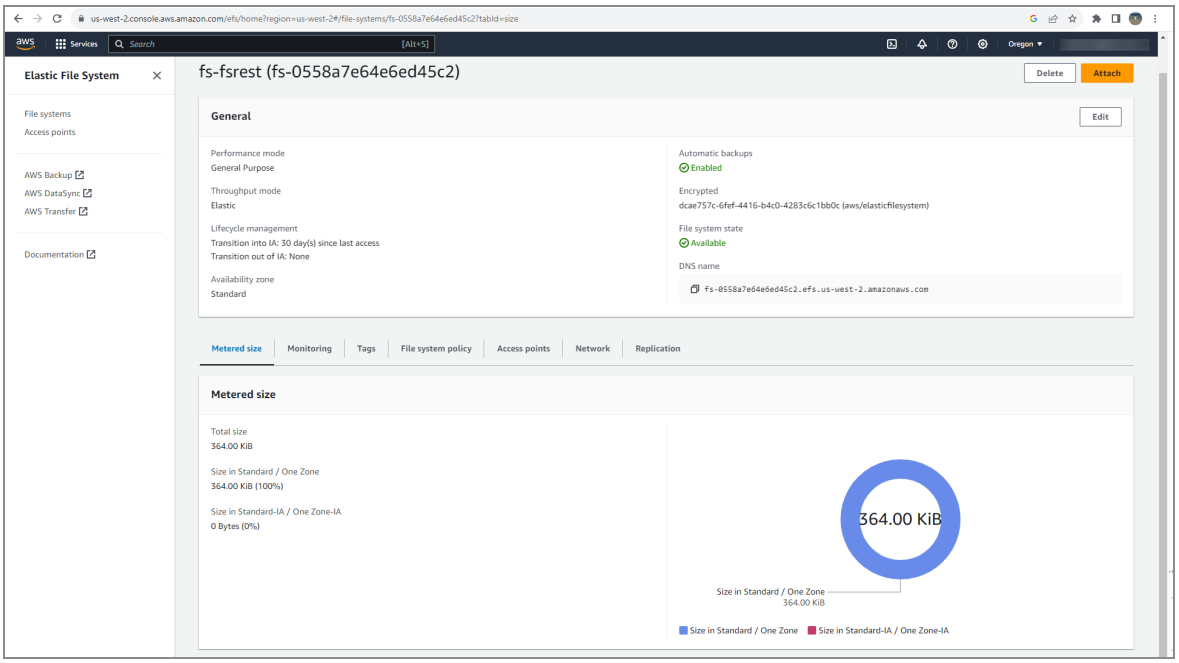
Note: If you opted to mount the Foresight REST API resources externally during the image build process, those resources can be used for mounting EFS on EC2 instance. For more information, see *TIBCO Foresight® REST API Installation*.

10. Create `FSRest.ini` in EFS folder and add all the properties.
11. Create docker.env, if required and add all the variables.
12. Run the Docker image:

```
sudo docker run --name fsrest --env-file=docker.env -p 8080:8080 -v
/home/ubuntu/efs:/usr/local/tomcat/TibcoFSRest -d
0.0.0.0:5000/fsrest:x.x.x.xxx
```

13. Upload files via a PUT request, the files are stored in EFS.

```
ubuntu@ip-xxx-xx-xx-xxx:~/efs$ ls
FSRest.ini  controlfiles  datafiles  guidelines  maps  test.txt
test2.txt
ubuntu@ip-xxx-xx-xx-xxx:~/efs$ cd datafiles/
ubuntu@ip-xxx-xx-xx-xxx:~/efs/datafiles$ ls
2Interchanges837i.txt  Covance_020212-20120216-132104-213.txt
ubuntu@ip-xxx-xx-xx-xxx:~/efs/datafiles$ ls -l
total 312
-rw-r----- 1 1100 2000      0 Oct 18 18:16 2Interchanges837i.txt
-rw-r----- 1 1100 2000 313230 Oct 18 18:17 Covance_020212-20120216-
132104-213.txt
```


us-west-2.console.aws.amazon.com/efs/home?region=us-west-2#/file-systems/fs-0558a7e64e6ed45c2?tabId=size
aws
Services
Search
[Alt+S]
Oregon
Elastic File System
File systems
Access points
AWS Backup
AWS DataSync
AWS Transfer
Documentation
fs-fsrest (fs-0558a7e64e6ed45c2)
Delete
Attach
General
Edit
Performance mode
General Purpose
Throughput mode
Elastic
Lifecycle management
Transition into IA: 30 day(s) since last access
Transition out of IA: None
Availability zone
Standard
Automatic backups
Enabled
Encrypted
dcae757c-6fef-4416-b4c0-4283c6c1bb0c (aws/elasticfilesystem)
File system state
Available
DNS name
fs-0558a7e64e6ed45c2.efs.us-west-2.amazonaws.com
Metered size
Monitoring
Tags
File system policy
Access points
Network
Replication
Metered size
Total size
364.00 KiB
Size in Standard / One Zone
364.00 KiB (100%)
Size in Standard-IA / One Zone-IA
0 Bytes (0%)
364.00 KiB
Size in Standard / One Zone
364.00 KiB
Size in Standard / One Zone
Size in Standard-IA / One Zone-IA

API Security

An API Key Security is a code used to authenticate an application. To enable API Key Security for Console and Silent Installation, see *TIBCO Foresight® REST API Installation Guide*.

The user needs authentication to use an API key security. To authenticate the user, send a GET request to the

```
<server-url>/ForesightREST/authenticate
```

Select **Authorization Type** as **Basic Auth**. Enter **Username** and **Password**.

By default, there are 2 users, one with admin scope and the other with register scope.

Users	Username	Password
User 1	admin	admin
User 2	fsrestclient	fsrestclient

The user 1 has the admin scope and user 2 has only the register scope. The list of scope is as follows:

Scope	Description
jobs/crd	The user can create, read and update jobs.
datafiles/urd	The user can read, update and delete datafiles.
guidelines/urd	The user can read, update and delete guidelines.
controlfiles/urd	The user can read, update and delete controlfiles.
outputfiles/rd	The user can read and delete outputfiles.

Scope	Description
fsenvvars/u	The user can update fsenvvars.
maps/urd	The user can update, read and delete maps.

The user with admin scope can create, read, update and delete all the scopes along with the registration of the application. Also, the user can assign scope as per the requirement, for example, to give permission for read only job, user can provide scope jobs/r.

The user with register scope can only register the application.

To register an application send a POST request to the

```
<server-url>/ForesightREST/register
```

Add the parameters mentioned in the below sample body:

```
{     "name": "job app",     "description": "Desc",     "organization":
"organization",     "scope": "jobs/crd outputfiles/rd datafiles/rd",
"emailId": "abc@pqr.com",     "contactName": "abc" }
```

Also, provide the API token or key received while authenticating the request.

Troubleshooting

This section lists some common errors and ways to troubleshoot them.

Issue	Resolution
Foresight REST API Configuration The request you have sent is valid, but the Foresight REST API is not configured properly. Sometimes the parameters are correct but some issues are found, such as Data Files are not present in the specified Directory path. In that case, HTTP response 200 Successful is sent to the submitter, and an error message is included in the body of the response.	Set RestDataPath correctly in `FSREST.ini`. This is applicable when the users deploy the war file directly to an application server, eg. Tomcat. The recommended method is to set `FSREST_DATADIR` in the system variable to configure Foresight REST API in container environment.

Check for Instream and Translation Engines
 	
During Startup
 	
If the Instream bin directory or Translator bin directory is not included in the system variable LD_LIBRARY_PATH(Unix), you receive an error message as an output in the console log.
 	
`$$$$$ FSRESTApp checking engines!`
`$$$$$ iStatus = -1 iVersion = 'Error: Can't find Instream C++ Library (jHVInStream.dll/.so)'`
 	
or
 	
`$$$$$ tStatus = -1 tVersion = 'Error: Can't find Translator C++ Library (jTranslator.dll/.so)'`
	Provide the file path for bin directory of Instream and Translator. Rerun `build-fsrest-engine.sh`.

Issue	Resolution

But the REST API engine starts.
 	
While POSTing Job Operation
 	
While POSTing a job operation, if the Instream bin directory or Translator bin directory is not included in the system PATH variable, you receive an error message as an output to POST response, with HTTP 400 Bad Request error.
 	
`The InStream engine is not found on the server.`
 	
or
 	
`The Translator engine is not found on the server.`
 	
If only one component is not found, POST operation for that component fails.
 	
For example: When Translator is not found, "IT" operation fails, but "I" operation runs.	

## Connection Errors You receive connection errors in the following cases: • When your certificate expires. • When your certificate is revoked by the certificate authority. • When the Foresight REST API application does not trust the certificate authority. • When the TLS version is mismatched. • When TOMCAT uses an outdated encryption algorithm	Follow the given steps for troubleshooting: • Replace the expired certificate with a new one. • Check with the authority to understand the reason for certificate revocation. • Replace your certificate with a trusted certificate.

Issue	Resolution
	• Avoid using an encryption algorithm which is not secure. Ensure that you use a secure encryption algorithm.

Appendix A - Guideline and Code Table Updates

Note: Foresight REST API does not support to apply Guidelines updates or Code table updates automatically, if the users need to update Guidelines/Code tables, they can do following manual steps. Changes made with `docker cp` or `kubectl cp` commands are limited to the running container or pod. For any reason if the container or pod restarts, the changes are lost.

For Guideline updates:

1. Decompress the package.
2. If the guidelines folder is mounted outside of the Kubernetes pod, copy the Database folder and static folder to the mounted guidelines folder.

   Or, if the guidelines folder is not mounted, compress the guideline files under the Database folder to a zip file, then upload the zip file using PUT `http://IP_ADDRESS:Port/ForesightREST/guidelines?multiple=y&overwrite=y`. Perform the same steps for the guidelines under the static folder.
3. Copy the files under the bin folder to Docker or Kubernetes containers with the following commands:

For the Docker:

```
docker cp bin/. Container_ID:/usr/share/tomcat/tibco/instream/fseng/bin/
```

For the Kubernetes:

```
kubectl cp bin/. namespace_name/pod_
name:/usr/share/tomcat/tibco/instream/fseng/bin/
```

For Code table updates:

1. Decompress the package.

2. Copy the `bin/fs_hipaa.dat` to Docker or Kubernetes containers with the following commands:

For the Docker:

```
docker cp bin/fs_hipaa.dat Container_
ID:/usr/share/tomcat/tibco/instream/fseng/bin/fs_hipaa.dat
```

For the Kubernetes:

```
kubectl cp bin/fs_hipaa.dat namespace_name/pod_
name:/usr/share/tomcat/tibco/instream/fseng/bin/fs_hipaa.dat
```

Appendix B - Integration with TIBCO Foresight Transaction Insight

Foresight REST API could send validation results file and the EDI file to the inbound or outbound folder of the Foresight Transaction Insight importer. To support the function, two new job POST parameters are added `publishReport` and `createTrk`. For more information, see Processing and Output Resources.

The following properties are set in the `FSREST.ini` file by default, for the directories to copy the EDI file and Details file. Foresight Transaction Insight importer should be able to access this location. Once a JOB request is done, the EDI file and details file get copied to the folders. The directories can be sub-folder under `mount_path` (for example, `/tmp`) set in the `standalone-deployment\config\deployment.properties` file.

`TIInboundPath` and `TIOutboundPath` can be set via system variables `FSREST_TIInboundPath` and `FSREST_TIOutboundPath`, with higher priority, and can be updated using PUT `updateEnv` request as well, see the following sections:

```
TIInboundPath=/tmp/inbound
```

```
TIOutboundPath=/tmp/outbound
```

For Inbound transactions, the B2B Gateway needs to send the files to the inbound folder defined by TIBCO.

For Outbound files, an internal application sends the files to the outbound folder defined by TIBCO.

The EDI data is renamed to `JobID.edi` and the details file is renamed to `JobID.dtl`. The EDI file can be passed via Data and InDatafileRef parameters. To skip one or multiple transaction set types for the copy operation, a new property `skipTS` is added in the `FSREST.ini` file, with the default value none, the valid values are none, or the transaction set names, separated by commas. For example, `837, 850`.

If you need to set the values, you can set the system variable `FSREST_SKIPTS`, or update the values with a PUT `updateEnv` request.

You can check the status of the copy operations in logs.

TIBCO Documentation and Support Services

For information about this product, you can read the documentation, contact Support, and join TIBCO Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the Product Documentation website, mainly in HTML and PDF formats.

The Product Documentation website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The documentation for this product is available on the TIBCO Foresight® REST API Product Documentation page.

How to Contact Support for TIBCO Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our product Support website.
- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the product Support website. If you do not have a username, you can request one by clicking **Register** on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature

requests from within the TIBCO Ideas Portal. For a free registration, go to TIBCO Community.

Legal and Third-Party Notices

Cloud SG software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the “readme” file for the availability of a specific version of Cloud SG software on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. CLOUD SG MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S), THE PROGRAM(S), AND/OR THE SERVICES DESCRIBED IN THIS DOCUMENT AT ANY TIME WITHOUT NOTICE.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "README" FILES.

This and other products of Cloud SG may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at https://www.cloud.com/legal.