COVIANT Software

USER GUIDE
DIPLOMAT MANAGED FILE TRANSFER
REST API
VERSION 8.2.1

V8.2.1

REST API USER GUIDE

Contacting Coviant Software LLC
Installation and configuration support is provided under warranty for 45 days from initial purchase, as well as under annual maintenance agreements. Email and phone support is available from 9 a.m. ET to 5 p.m. ET weekdays. If you require assistance, contact Coviant Software support as follows:

Voice: 781.210.3310 x2

Fax: 781.210.3313

Web: www.coviantsoftware.com

E-mail: support@coviantsoftware.com

Table of Contents

1 Overview

The Diplomat MFT REST API enables development and/or extension of third-party applications to run file transfer jobs and obtain job status from the Diplomat MFT Service. Use of the REST API requires a license that allows requests from the API.

The REST API is based on HTTP or HTTPS version 1.1. All messaging is synchronous request/response and enables client processes to make requests to the Diplomat MFT Service. Using API requests, you can:

- Run existing jobs
- Run ad-hoc jobs
- Cancel jobs
- Terminate jobs
- Obtain current job status
- Obtain job summaries
- Obtain job history, including log entries

The Diplomat MFT REST API adheres to the following points of the REST paradigm:

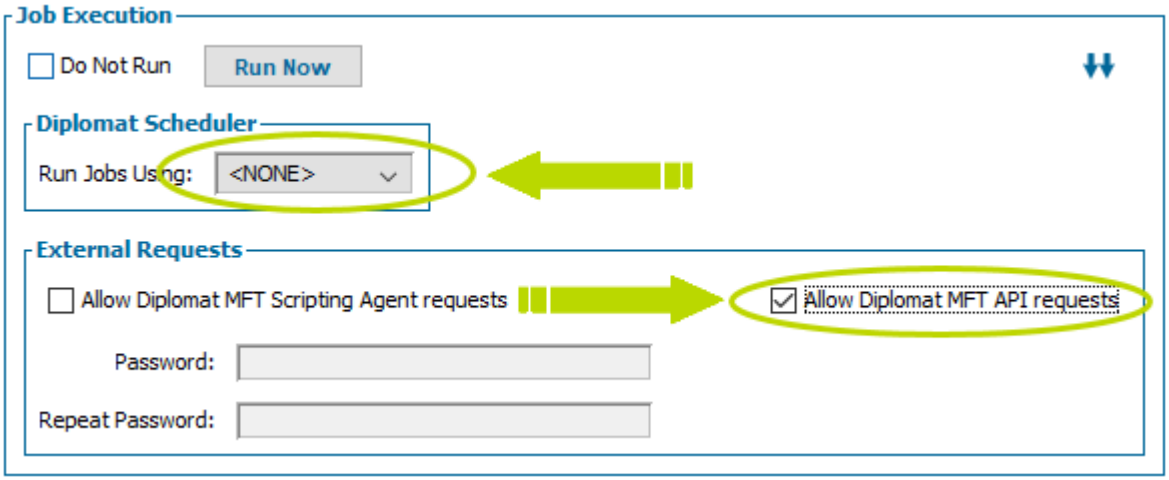
- Makes use of a uniform interface that enables a separation of concerns between client and server.
- Enables stateless client-server interaction, except with regard to authentication. After a client is authenticated, the session information is persisted via the use of standard HTTP cookies.
- Uses the standard HTTP protocol, using HTTP verbs and not requiring the use of any other specific protocol or language.
- Allows for client platform and language independence (i.e. the client software can run on any operating system and can be developed in any language).
- Presents response data using standard XML syntax and, in some cases, JSON.

2 Create Target Transaction

To use the *Run Now* API endpoint, a target transaction must be defined in Diplomat MFT. For *Run Now Ad-Hoc* requests, the request itself will contain the definition of the job to execute.

Open the Diplomat MFT Client. Create an inbound or outbound transaction with the desired characteristics to be executed using the API. *Check Allow Diplomat MFT API requests* on the *Job Execution* panel on the transaction screen.

If you want to ONLY allow Diplomat MFT API requests, then set *Run Jobs Using* in the Job Execution panel to <NONE>. **Do not** check the *Do Not Run* box, because doing so will prevent execution by the REST API call.



3 API Requests/Response Codes

The REST API uses request/response messaging. The protocol for requests/responses is HTTP or HTTP/S version 1.1.

The Diplomat MFT Service, by default, listens on port 8080 for HTTPS connections and on port 8443 for HTTP connections. API requests should be directed to the URI /diplomat/APIRequest as POST requests. **NOTE:** If the Diplomat MFT Service is configured to listen on other ports (e.g., for Diplomat MFT Client or Diplomat MFT Job Monitor connections), then those ports will need to be used with the REST API connections, as well.

Connections with the Diplomat MFT Service are initiated by establishing a session on the Diplomat MFT Service using a Connect request. If a session is established successfully, a cookie identifying the session is returned to the requesting client process. This cookie must be included in any subsequent API requests –up to and including the *Disconnect* request to end the session. **NOTE:** Sessions are automatically ended if the Diplomat MFT Service senses no client process interaction during the session expiration time specified under Settings > Session Management in the Diplomat MFT Client.

Requests are transmitted as name/value pairs in the query portion of the HTTP URI, which must be URL encoded. One name/value pair in a request always indicates the type of request being made (e.g., run a job now). The other name/value pairs contain parameters for the particular request (e.g., the ID of the job to be run).

Responses to an API request are returned as HTTP responses. The response follows the standard HTTP response protocol which contains a response code followed by the response body.

- The *Response Code* indicates in general whether or not the request was successful.
- The *Response Body* returns the information requested from the Diplomat MFT Service. The response body may be blank if the response code is self-explanatory or it may contain content, which is XML formatted. The XML details of responses that contain significant content are detailed in the Appendix.

3.1 Standard Response Codes

All HTTP/S API *request*s can result in the following response codes, unless otherwise noted:

- 200 – Successful
- 400 – Bad request – one or more parameters are missing or invalid
- 401 – Unauthorized – an active session was not found or could not be established
- 500 – Internal server error - Diplomat MFT Service error
- 501 – Not implemented - API is not supported by the Diplomat MFT Service

If the response code is 400 or 500, the response body typically contains a description of the specific error.

3.2 Requests

The REST API supports the following requests. The form of each request is defined below. All parameters are required unless otherwise noted. Additional response codes for specific requests are also shown.

3.2.1 Cancel Job Request

The *Cancel* request is used to cancel an executing job. If successful, it results in a "cancel" signal being sent to the Diplomat job. After receiving the "cancel" request, the Diplomat MFT Service attempts to end the job gracefully. **NOTE:** To cause a job to end more forcefully, see the *Terminate Job* request section below.

Request parameters are:

- Name = "requestType", Value = "cancel"
- Name = "transactionID", Value = Transaction ID of the job to be canceled
- Name = "transactionName", Value = Transaction name of the job to be canceled.

NOTE: Either a transactionID or transactionName must be supplied but both are not required. If both are supplied, transactionName is ignored.

A response code of 400 indicates that there is no running job for the given transactionID/transactionName.

3.2.2 Connect Request

The *Connect* request is used to establish a session with the Diplomat MFT Service. If successful, a cookie is returned in the response header. This cookie identifies the session and must be included in any subsequent API requests. **NOTE:** A *Connect* request must be successful before any other request can be processed successfully.

Request parameters are:

- Name = "requestType", Value = "connect"
- Name = "domain", Value = domain of the client process (Optional)
- Name = "userID", Value = userID of the client process
  The *domain* and *userID* parameters are used to identify the client process. If the domain parameter is included, the client process identification is domain/userID. If the domain parameter is not included, the client process identification is simply userID. All requests associated with the session established by this request are tagged in the Diplomat log with the client process identification. **NOTE:** The client process identification is used only to identify the client process in log entries. It is not used for authentication or authorization.
- Name = "userName", Value = userName used for authentication
- Name = "password", Value = password used for authentication
  The *userName/password* combination must match one of the userName/password combinations defined in the Diplomat User DB.
- Name = "disconnect", Value =the connectionID of the user to disconnect (Optional)
  If the *disconnect* parameter is provided, its value identifies a current client process that should be disconnected to allow this connection. This parameter is typically used in a request resubmission after the original submission resulted in a 409 response code. The connectionID is returned in the 409 response code.

A response code of 401 indicates that the username/password combination supplied is not valid.

Additional possible response codes:

- 409 – Maximum number of concurrent sessions are already in use. The response body should contain a list of the user identifications of the currently connected clients. **NOTE:** The request can be resubmitted indicating the connectionID of the client that should be disconnected.

3.2.3 Disconnect Request

The *Disconnect* request is used to end a session with the Diplomat MFT Service. After the *Disconnect* request is processed, no other request will be successful until a successful *Connect* request establishes a new session.

Request parameters are:

- Name = "requestType", Value = "disconnect"

3.2.4 Job Histories Request

The *Job Histories* request is used to obtain the job history records associated with the specified Transactions.

Request parameters are:

- Name = "requestType", Value = "jobHistories"
- Name = "transactionID", Value = ID of a transaction whose job history is requested.
- Name = "transactionName", Value = Name of a transaction whose job history is requested.

NOTE: The transactionID and/or transactionName parameters may be repeated to obtain the job histories of multiple transactions.

Additional possible response code:

- 409 – Job histories not supported by Diplomat MFT Service

If the response code is 200, the response body contains the requested history records.

NOTE: If no job history records exist for a given transactionID or transactionName, the response will have an empty list of history records for the corresponding transaction.

3.2.5 Job History Request

The *Job History* request is used to obtain the job history record associated with the job identified by the given sequence number.

NOTE: The sequence number is the return value of a previously executed *RunNow Job Request*.

Request parameters are:

- Name = "requestType", Value = "jobHistory"
- Name = "sequence", Value = sequence number of the job whose history is requested.

If the response code is 200, the response body contains the requested history record.

3.2.6 Job Summary Request

The *Job Summary* request is used to obtain the textual job summary of the most recent job associated with the given transaction ID.

Request parameters are:

- Name = "requestType", Value = "summary"
- Name = "transactionID", Value = ID of a transaction whose job summary is requested.
- Name = "transactionName", Value = Name of a transaction whose job summary is requested.

NOTE: Either a transactionID or transactionName must be supplied but both are not required. If both are supplied, transactionName is ignored.

If the response code is 200, the response body contains the job summary. A response code of 400 can occur when no job summary records match the transactionID/transactionName.

3.2.7 Log Buffer Request

The *Log Buffer* request is used to access a portion of a Log View. **NOTE:** See *Log View Request* section below for a general discussion of how to view a log via the API.

Request parameters are:

- Name = "requestType", Value = "logBuffer"
- Name = "logViewID", Value = ID of the log view
- Name = "lineNum", Value = line number – zero-based and with respect to the entire log view, i.e., 0 to n-1.

If the response code is 200, the response body contains the log buffer which contains the line of text specified by the lineNum parameter, assuming the lineNum parameter specifies a valid line number for the given log view. If the lineNum parameter is less than zero, the view's first buffer is returned. If the lineNum parameter is not less than the size of the view, the view's last buffer is returned. A response code of 400 can occur when no log view can be found with the requested logViewID.

3.2.8 Log View Request

The *Log View* request is used to define a log view for a particular job and, optionally, for a given textual search string. To view a log via the API the following sequence of requests must be made:

1. Determine the sequence number that identifies the job in the Job History DB. Generally, this is done by making a *Job Histories* request and processing the response.
2. Use a *Log View* request to define a log view.
3. Use a series of *Log Buffer* requests to access specific lines from the log view.
4. Delete the log view via a *Remove Log View* request. **NOTE:** It is very important to always delete a log view when it is no longer needed to free up space on the system running the Diplomat MFT Service

Request parameters are:

- Name = "requestType", Value = "logView"
- Name = "transactionName", Value = Name of the transaction that corresponds to the job.
- Name = "sequence", Value = sequence number which identifies the job in the Job History DB.
- Name = "searchText", Value = text which log records must contain to be included in the view (Optional).

Additional possible response code:

404 – No job history records exist for the given transactionName and sequence number combination or no log files can be found that contain records for the given parameters.

If the response code is 200, the response body contains the log view ID, the total number of lines of text in the view and the number of lines that will be returned in each log buffer.

3.2.9 Remove Log View Request

The *Remove Log View* request is used to delete a log view. See *Log View Request* section above for a general discussion of how to view a log via the API.

Request parameters are:

- Name = "requestType", Value = "removeLogView"
- Name = "logViewID", Value = ID of the log view

A response code of 400 indicates that no log views match the log view ID.

3.2.10 RunNow Job Request

The *RunNow Job* request is used to schedule a job for ASAP execution. If successful, the job has been scheduled for execution.

Request parameters are:

- Name = "requestType", Value = "runNow"
- Name = "transactionID", Value = Database ID of the transaction for which a job is to be run.
- Name = "transactionName", Value = Name of the transaction for which a job is to be run.
- Name = "var", Value = "variable_name|value" where variable_name and value represent the variable's name and the value used to replace the <%variable_name%> string in the source or destination file(s) field(s) of the transaction at run time. **NOTE**: Multiple "var" parameters can be used in a single request.

NOTE: Either a transactionID or transactionName must be supplied but both are not required. If both are supplied, transactionName is ignored.

A response code of 400 indicates that the requested job cannot be scheduled.

3.2.11 RunAdHoc Job Request

The *RunAdHoc Job* request is used to schedule an ad-hoc job between two **partners** for ASAP execution. This request does **not** require a pre-existing transaction to be defined in Diplomat MFT; instead, the *RunAdHoc Job* request requires a **source** and **destination** partner to be defined within Diplomat MFT, and the rest of the request payload supplies all the details of the job to be executed. If successful, the job has been scheduled for execution.

This REST API endpoint accepts parameters as query strings on the URL, ***or*** as a JSON payload. Be sure to set the "Content-Type" header of your request to "application/json" when submitting a JSON payload.

Request parameters when using the QueryString method of invocation are:

Name	Value
requestType	runAdHoc
sourcePartnerName	name of the **source** Partner defined in Diplomat MFT. *(required)*
destinationPartnerName	name of the **destination** Partner defined in Diplomat MFT *(required)*
sourceFileName	The filename to match at source partner for transferring to destination partners. This is equivalent to "Source File Info" in the UI. *(required)*
Synchronous	If this parameter is present, the job is executed synchronously (meaning that it runs through to completion before an HTTP response is returned). If this parameter is absent, the job executes in the same fashion as "runNow", which immediately returns a sequence number that can be used for querying status. **This parameter can be used for JSON payload requests as well**.

When supplying parameters via a JSON payload to the request, these is the structure of the parameters:

Minimal Required Configuration:

These represent the **required** parameters to the RunAdHoc endpoint.:

```
{
  "sourcePartnerName" : "laptopA",
  "destinationPartnerName" : "laptopZ",
  "FileInfo" : [
      {
          "sourceFileName" : "*.pdf"
   ]
```

```
}
```

"Advanced" Configuration options with example values:

These represent the **complete set** of parameters that can be passed to the RunAdHoc endpoint. Leave any parameter out and the default value will be supplied.

```
{
 "direction" : "Outbound ",
 "sourcePartnerName" : "Coviant SFTP",
 "destinationPartnerName" : "Big Community Bank",
 "FileInfo" : [
     {
        "sourceFileName" : "*.pdf",
        "destinationFileName" : "<DATE>/|",
        "useModifyDate" : "true",
        "numberOfFiles" : 0,
        "required" : "false",
        "ignoreFileHandling" : "false"
 ],
 "sourceDateFormat" : "",
 "sourceDateRange" : "Today",
 "modifyDateRange" : "Today",
 "destinationDateFormat" : "<YYYY><MM><DD>",
 "destinationDate" : "SourceDate",
 "overwrite" : " Overwrite if source newer or different size",
 "removeSourceDate"  : "false",
 "removeSourceSequence"  : "false",
 "allowZeroByteFiles"  :"true",
 "deleteSource" : "false",
 "moveSource" : "true",
 "moveSourceSuccessDestination" : "/processed",
 "moveSourceErrorDestination" :  "/error",
 "transferOrder" : "Oldest First",
 "failIfFilesNotFound" : "true",
 "translateInvalidWindowsChars" : "true",
 "fileHandling" : {
    "encrypt" : "true",
    "sign" : "true",
    "onePassSignature" : "true",
    "asciiArmoring" : "true",
    "compress" : "true",
    "canonicalText" : "true",
    "integrityProtectPacket" : "true"
 },
 "emailNotifications" : {
    "businessRecipients" : [
       {
          "email" : "marketing@coviantsoftware.com",
          "notificationType" : "All"
       },
       {
          "email" : "support@coviantsoftware.com",
          "notificationType" : "Warning and Failure"
       }
    ]
 }
 "skipPrimaryArchiving" : "true",
 "additionalLocation" : "\\FS001\DiplomatArchive\",
 "zipArchiveFiles" : "true",
 "fileTypes" : "Source",
 "archiveTriggerType" : "SuccessWarning",
```

```
   "skipPrimaryArchiving" : "true",
   "advancedTroubleShooting" : "false",
   "zip" : {
      "zipType" : "ZIP",
      "zipStyle" : "Together",
      "zipFilesLocation" : "\\FS001\DiplomatData\Outbound\FOO",
      "zipFilesPattern" : "*.*",
      "zipPath" : "\\FS001\DiplomatData\Outbound\FOO\ZIPPED\<DATE>",
      "zipDateFormat" : "<YYYY><MM><DD>.<HH><mm><ss>",
      "deleteUnzippedFiles" : "true",
      "zipEncryption" : "true",
      "zipPassword" : "SecretPassword",
      "continueOnZipFailure" : "false"
   }
}
```

The above represents an OUTBOUND transaction, and the "zip" object displays the available options for outbound ZIP actions. Below is what an INBOUND transaction provides as options for the "zip" object:

```
   "unzip" : {
      "zipType" : "GZIP",
      "zippedFilesPattern" : "*.tar.gz",
      "unzipFolderPaths" : ["\\FS001\DiplomatData\Outbound\FOO\UNZIPPED\<DATE>",
      "zipDateFormat" : "",
      "zipPassword" : "SecretPassword",
      "unzipOverwrite" : "Overwrite",
      "deleteZippedFiles" : "true",
      "executeUnzipOnFailure" : "false"
   }
```

3.2.12 Terminate Job Request

The *Terminate Job* request is used to terminate an executing job. If successful, it results in a "terminate" signal being sent to the job, after which, the job's thread is forcefully terminated. To cause a job to end more gracefully, see the *Cancel Job Request* section above.

Request parameters are:

- Name = "requestType", Value = "terminate"
- Name = "transactionID", Value = Transaction ID of the job to be terminated
- Name = "transactionName", Value = Transaction name of the job to be canceled.

NOTE: Either a transactionID or transactionName must be supplied but both are not required. If both are supplied, transactionName is ignored.

A response code of 400 indicates that there is no running job for the given transactionID/transactionName.

3.2.13 Transaction Statuses Request

The *Transaction Statuses* request is used to obtain the status of all transactions or just those transactions whose status has changed since a given time.

Request parameters are:

- Name = "requestType", Value = "transactionStatuses"
- Name = "since", Value = time value (Optional)

The format of the "since" parameter is milliseconds since January 1 1970 12 AM. If the "since" parameter is provided, only transactions whose status changed after the specified time are included in the response.

If the response code is 200, the response body contains the time the data was collected and a collection of transaction statuses.

4 API Usage Example

This section describes the interactions between an API client and the Diplomat MFT Service to run a Diplomat job and to obtain the job's completion status. The requests described below can be entered typed into the address line of a web browser and the responses would be displayed in the browser's content area.

The following assumptions have been made with regard to this sample:

- The target Diplomat MFT Service runs on a system identified as 'diplomatServer' and listens on port 8080 for secure HTTP connections.
- The Diplomat database contains a user account with the username 'Administrator' and the password 'diplomat123'.
- The Diplomat database contains a transaction with Transaction ID ‘outboundSample'.
- If using a browser, the browser is set to allow cookies.

4.1 Connect

Before any other interactions can take place with the Diplomat MFT server, a Connect request must be sent to establish a valid session. The request to connect using the credentials of the user identified above is:

https://diplomatServer:8080/diplomat/APIRequest?requestType=connect&userID=Jim.Smith&userName=Administrator&password=diplomat123

The response from the Diplomat MFT Service is

```
<diplomatApiResponse responseType="connectResponse">
        <statusCode>200</statusCode>
        <applVersion>6.1.2</applVersion>
        <serverBuildID>20160211</serverBuildID>
        <serverDisplayName>CoviantLaptop</serverDisplayName>
        <diplomatUser>
                <id>0000000000000001</id>
                <contactName>Administrator</contactName>
                <userRole>Administrator</userRole>
                <userName>Administrator</userName>
                <lastPasswordUpdate>1444918078230</lastPasswordUpdate>
                <domain></domain>
                <userID></userID>
                <twoFactorAuth>false</twoFactorAuth>
                <updatedToV52>true</updatedToV52>
        </diplomatUser>
        <authStrategy>Password</authStrategy>
        <maxConnections>false</maxConnections>
        <passwordExpired>false</passwordExpired>
        <noSession>false</noSession>
</diplomatApiResponse>
```

The status code ‘200’ signals that:

- The connection was successfully authenticated and a session has been initiated.
- A session cookie has been returned as an HTTP header. If you are using a browser, it automatically sends that cookie as an HTTP header in any subsequent requests to the Diplomat MFT Service. If you are using

custom software, then the cookie information must be included in all subsequent requests during the session.

4.2 RunNow Job

The request to run the job 'outboundSample' is:

https://diplomatServer:8080/diplomat/APIRequest?requestType=runNow&transactionName=outboundSample&var=filename|301&var=ext|txt

The response from the Diplomat MFT Service is:

```
<diplomatApiResponse responseType="runNowResponse">
        <statusCode>200</statusCode>
        <sequence>0000000000123456</sequence>
</diplomatApiResponse>
```

The status code '200' signals that the request was successful and the Diplomat MFT Service was able to schedule the 'outboundSample' job for execution.

NOTE: Successful scheduling of the job does NOT indicate whether the job run successfully. *Job History* requests must be used with the returned sequence number as a parameter to determine whether the job ran successfully.

4.3 RunAdHoc Job

4.3.1 URL Querystring Parameter-based invocation (asynchronous)

The request to run an ad-hoc job between the source partner 'Mainframe" and the destination partner "AcmeBank" is:

https://diplomatServer:8080/diplomat/APIRequest?requestType=runAdHoc&sourcePartnerName=Mainframe&destinationPartnerName=AcmeBank&sourceFileName=*.pdf

The response from the Diplomat MFT Service is:

```
<diplomatApiResponse responseType="runAdHocResponse">
        <statusCode>200</statusCode>
        <sequence>0000000000123456</sequence>
</diplomatApiResponse>
```

The status code '200' signals that the request was successful and the Diplomat MFT Service was able to schedule the 'outboundSample' job for execution.

NOTE: Successful scheduling of the job does NOT indicate whether the job run successfully. *Job History* requests must be used with the returned sequence number as a parameter to determine whether the job ran successfully.

4.3.2 JSON Parameter-based invocation (synchronous)

The request to run an ad-hoc job between the source partner 'Mainframe" and the destination partner "AcmeBank", using a JSON payload and forcing synchronous execution is:

https://diplomatServer:8080/diplomat/APIRequest?requestType=runAdHoc&synchronous

with a "Content-Type" header of "application/json" and a POST body of:

```
{
  "direction" : "Outbound",
  "sourcePartnerName" : "Mainframe",
  "destinationPartnerName" : "AcmeBank",
  "FileInfo" : [
      {
         "sourceFileName" : "*.pdf",
         "destinationFileName" : "<DATE>/|"
   ],
   "destinationDateFormat" : "<YYYY><MM><DD>",
   "deleteSource" : "true",
   "fileHandling" : {
      "encrypt" : "true",
      "sign" : "true",
      "compress" : "true"
   },
   "emailNotifications" : {
      "businessRecipients" : [
         {
            "email" : "info@coviantsoftware.com",
            "notificationType" : "All"
         },
         {
            "email" : "support@coviantsoftware.com",
            "notificationType" : "Warning and Failure"
         }
      ]
   }
   "skipPrimaryArchiving" : "true"
}
```

The JSON response from the Diplomat MFT Service is:

```
{"diplomatApiResponse": {
   "sequence": "0000000000000046",
   "responseType": "runAdHocResponse",
   "jobHistory": {
      "sequence": "0000000000000046",
      "completion": 1,
      "fileHistory": [
         {
            "sequence": "0000000000000046",
            "size": 22757,
            "numRetries": 0,
            "destination": "rpt1848174453034255861.pdf",
            "startTime": 1574795962046,
            "source": "rpt1848174453034255861.pdf",
            "endTime": 1574795962163,
            "completionStatus": "Warning",
            "status": "Complete"
         },
         {
            "sequence": "0000000000000046",
            "size": 1540,
            "numRetries": 0,
            "destination": "rpt4691868228510628466.pdf",
            "startTime": 1574795962169,
            "source": "rpt4691868228510628466.pdf",
            "endTime": 1574795962206,
            "completionStatus": "Warning",
            "status": "Complete"
         },
```

```
         {
            "sequence": "0000000000000046",
            "size": 33010,
            "numRetries": 0,
            "destination": "rpt6514926645313741279.pdf",
            "startTime": 1574795962212,
            "source": "rpt6514926645313741279.pdf",
            "endTime": 1574795962245,
            "completionStatus": "Warning",
            "status": "Complete"
         },
         {
            "sequence": "0000000000000046",
            "size": 58664,
            "numRetries": 0,
            "destination": "rpt7367590745193729909.pdf",
            "startTime": 1574795962251,
            "source": "rpt7367590745193729909.pdf",
            "endTime": 1574795962285,
            "completionStatus": "Warning",
            "status": "Complete"
         }
      ],
      "filesSuccessful": 0,
      "filesFailed": 4,
      "execAttempt": 1,
      "startTime": 1574795961904,
      "endTime": 1574795962341,
      "completionStatus": "Warning",
      "transactionName": "AdHoc_Mainframe_AcmeBank_20191126191921",
      "status": "Complete",
      "filesFound": 4
   },
   "transactionName": "AdHoc_Mainframe_AcmeBank_20191126191921",
   "statusCode": 200
}}
```


4.4 ListPartners Request

The request to list all partners defined in Diplomat MFT is:

https://diplomatServer:8080/diplomat/APIRequest?requestType=listPartners

Responses can be delivered in XML (default) or JSON; to obtain a JSON response, be sure to include an "Accept: application/json" header in the request, or include the parameter name "json" in the querystring.

"Anonymous" partners are those which are defined within a transaction itself (using Source Partner "<NONE>").

The JSON response from the Diplomat MFT Service is:


```
{"diplomatApiResponse": {
   "responseType": "listPartnersResponse",
   "partners": {"partner": [
      {
         "name": "AcmeBank",
         "serverAddress": "",
         "description": "",
         "anonymous": false,
         "transportType": "Local Network",
         "partnerType": "Public"
```

```
      },
      {
         "name": "Mainframe",
         "serverAddress": "",
         "description": "",
         "anonymous": false,
         "transportType": "Local Network",
         "partnerType": "Trusted"
      },
      {
         "name": "",
         "serverAddress": "",
         "description": "",
         "anonymous": true,
         "transportType": "Amazon S3",
         "partnerType": "Public"
      }
  ]},
   "statusCode": 200
}}
```


The XML response looks like:


```
<diplomatApiResponse responseType="listPartnersResponse">
  <statusCode>200</statusCode>
  <partners>
    <partner>
      <name>AcmeBank</name>
      <description></description>
      <anonymous>false</anonymous>
      <partnerType>Public</partnerType>
      <transportType>Local Network</transportType>
      <serverAddress></serverAddress>
    </partner>
    <partner>
      <name>Mainframe</name>
      <description></description>
      <anonymous>false</anonymous>
      <partnerType>Trusted</partnerType>
      <transportType>Local Network</transportType>
      <serverAddress></serverAddress>
    </partner>
    <partner>
      <name></name>
      <description></description>
      <anonymous>true</anonymous>
      <partnerType>Public</partnerType>
      <transportType>Amazon S3</transportType>
      <serverAddress></serverAddress>
    </partner>
  </partners>
</diplomatApiResponse>
```


4.5 Transaction Statuses

The request to get transaction statuses is:

https://diplomatServer:8080/diplomat/APIRequest?requestType=transactionStatuses&since=1458583582353

The 'since' parameter, which represents a time value, expressed in units used internally by the Diplomat MFT Service is optional. When used, it limits the response to only transactions whose status has changed since the given time. In this case, we have used the value that was returned by our *RunJob r*equest as the scheduled time of the job we want to monitor.

NOTE: Each transaction statuses response contains the JobHistory record of the most recent associated job, if there is one.

The response from the Diplomat MFT Service is:

```
<diplomatApiResponse responseType="transactionStatusesResponse">
    <statusCode>200</statusCode>
    <monitorTime>1458584289309</monitorTime>
    <dbUpdating>false</dbUpdating>
    <transactionStatusSummary row="0" column="0" value="2"/>
    <transactionStatusSummary row="0" column="1" value="4"/>
    <transactionStatusSummary row="1" column="0" value="1"/>
    <transactionStatusSummary row="1" column="1" value="0"/>
    <transactionStatusSummary row="2" column="0" value="0"/>
    <transactionStatusSummary row="2" column="1" value="0"/>
    <transactionStatusSummary row="3" column="0" value="0"/>
    <transactionStatusSummary row="3" column="1" value="0"/>
    <transactionStatus>
        <monitorTime>1458584289397</monitorTime>
        <transactionId>0000000000000010</transactionId>
        <transactionName>outboundSample</transactionName>
        <transactionType>Outbound</transactionType>
        <transactionState>Not Scheduled</transactionState>
        <jobHistory>
            <sequence>0000000000000092</sequence>
            <transactionName>outboundSample</transactionName>
            <startTime>1458583582353</startTime>
            <endTime>1458583588192</endTime>
            <status>Complete</status>
            <completionStatus>Successful</completionStatus>
            <filesFound>1</filesFound>
            <filesSuccessful>1</filesSuccessful>
            <filesFailed>0</filesFailed>
            <execAttempt>1</execAttempt>
            <completion>1.0</completion>
            <fileHistory>
                <sequence>0000000000000092</sequence>
                <source>file1.txt</source>
                <destination>file1.txt</destination>
                <startTime>1458583588103</startTime>
                <endTime>1458583588182</endTime>
                <status>Complete</status>
                <completionStatus>Successful</completionStatus>
                <size>7</size>
                <numRetries>0</numRetries>
            </fileHistory>
        </jobHistory>
    </transactionStatus>
```

</diplomatApiResponse>

The status code of 200 indicates that the Transaction Statuses request was successful.

The pertinent data, for the purposes of this example, is contained in the <status> tag in the JobHistory portion of the XML bolded above. Valid values are:

- **Delayed** Job executed using a Diplomat MFT Scripting Agent command with a <delay> parameter.
- **Queued** Job waiting in queue for execution.
- **Running** Job actively executing.
- **Cancelling** Request to cancel job occurred, but job is still executing.
- **Terminating** Request to terminate job occurred, but job is still executing.
- **Aborting** Unrecoverable error encountered.

Transaction Statuses requests can be issued on a polling basis, if necessary, in order to keep the client session alive until the job is complete.

Once the job is complete the overall job completion status is displayed in the <completionStatus> tag in the JobHistory portion of the XML bolded above. Valid values are:

- **Preview License** — Job attempted to run, but was stopped due to no valid license.
- **File(s) not Found** — No files were found on the most recent execution and transaction was **not** set to Fail if File(s) Not Found.
- **Required File(s) Not Found** — Some files were found on the most recent execution, but one or more required files were **not** found and transaction was **not** set to Fail if File(s) Not Found.
- **Successful** — Most recent execution completed successfully. Email and other notifications indicate the job was *Successfu*l.
- **Warning** — Job completed successfully, but had at least one error that might have affected the integrity of the file(s) being transferred.

  Examples of problems generating a Warning status, include:
  - Error closing a file
  - Error deleting an uploaded file after a problem during transmission
  - Decryption or verification key is not valid for current date
  - ASCII file size not within tolerance

  NOTE: Source files with a Warning status are NOT deleted.
- **Failure** — Most recent job execution did not complete successfully. Email and other notifications indicate the job was *Failure*.
- **Critical** — Job failed due to a problem with the audit database and audit trail settings set to T*reat Failures as Critical*.
- **Cancelled** — Job manually cancelled on most recent execution.
- **Terminated** — Job manually terminated on most recent execution.
- **Incomplete** — Diplomat MFT Service stopped during job execution.
- **Missed** — Jobs are flagged as missed, when Fail if File(s) Not Found is checked and:
  - Diplomat MFT Service not running when last job execution scheduled.
  - Job queued, but execution had not started, when Diplomat MFT Service stopped unexpectedly.

4.6 Disconnect

A *Disconnect* request should always be sent to the Diplomat MFT Service to close the session.

The request to disconnect is:

https://diplomatServer:8080/diplomat/APIRequest?requestType=disconnect

The response from the Diplomat MFT Service is:

```
<diplomatApiResponse responseType="okResponse">
        <statusCode>200</statusCode>
</diplomatApiResponse>
```

The status code of 200 indicates that the *Disconnect* request was successful.

5 Appendix: XML Responses

The body of each HTTP response may be blank if the response code is self-explanatory or it may contain content, which is XML formatted. Examples of the XML details of responses that contain significant content are provided below.

5.1 Connect Request Response

5.1.1 200 Response Code

```
<diplomatApiResponse responseType="connectResponse">
    <statusCode>200</statusCode>
    <applVersion>6.1.2</applVersion>
    <serverBuildID>20160211</serverBuildID>
    <serverDisplayName>CoviantLaptop</serverDisplayName>
    <diplomatUser>
        <id>0000000000000001</id>
        <contactName>Administrator</contactName>
        <userRole>Administrator</userRole>
        <userName>Administrator</userName>
        <lastPasswordUpdate>1444918078230</lastPasswordUpdate>
        <domain></domain>
        <userID></userID>
        <twoFactorAuth>false</twoFactorAuth>
        <updatedToV52>true</updatedToV52>
    </diplomatUser>
    <authStrategy>Password</authStrategy>
    <maxConnections>false</maxConnections>
    <passwordExpired>false</passwordExpired>
    <noSession>false</noSession>
</diplomatApiResponse>
```

5.1.2 409 Response Code

```
<diplomatApiResponse responseType="connectResponse">
    <statusCode>409</statusCode>
    <applVersion>6.1.2</applVersion>
    <serverBuildID>20160211</serverBuildID>
    <serverDisplayName>CoviantLaptop</serverDisplayName>
    <maxConnections>true</maxConnections>
    <diplomatConnection>
        <connectionID>
            6f24d2d5-3401-42ec-a0b3-ce6d313b10dc
        </connectionID>
        <sessionIpaddress>127.0.0.1</sessionIpaddress>
        <sessionDomain></sessionDomain>
        <sessionUserID>Jim</sessionUserID>
        <contactName>Administrator</contactName>
    </diplomatConnection>
    <passwordExpired>false</passwordExpired>
    <noSession>false</noSession>
</diplomatApiResponse>
```

5.2 Job Histories Request Response

5.2.1 200 Response Code

```
<diplomatApiResponse responseType="jobHistoriesResponse">
    <statusCode>200</statusCode>
    <jobHistoryEnabled>true</jobHistoryEnabled>
    <jobHistoryClientUpdateRate>10</jobHistoryClientUpdateRate>
    <jobHistories>
            <monitorTime>1457970974435</monitorTime>
            <transactionName>inftpsold</transactionName>
            <jobHistory>
                    <sequence>0000000000000083</sequence>
                    <updateTime>1457970273040</updateTime>
                    <transactionName>inftpsold</transactionName>
                    <startTime>1451771368454</startTime>
                    <endTime>1451771442559</endTime>
                    <status>Complete</status>
                    <completionStatus>Successful</completionStatus>
                    <filesFound>1</filesFound>
                    <filesSuccessful>1</filesSuccessful>
                    <filesFailed>0</filesFailed>
                    <execAttempt>1</execAttempt>
                    <completion>1.0</completion>
                    <fileHistory>
                                <sequence>0000000000000083</sequence>
                                <source>big.txt</source>
                                <destination>big.txt</destination>
                                <startTime>1451771371425</startTime>
                                <endTime>1451771385325</endTime>
                                <status>Complete</status>
                                <completionStatus>Successful</completionStatus>
                                <size>39083371</size>
                                <numRetries>0</numRetries>
                    </fileHistory>
            </jobHistory>
    </jobHistories>
</diplomatApiResponse>
```

5.3 Job Summary Request Response

5.3.1 200 Response Code

```
<diplomatApiResponse responseType="summaryResponse">
    <statusCode>200</statusCode>
    <summary>Outbound transaction

Source files obtained from C:/downloads/thru/demo
    FilesComplete    Last modified: 20151217.110832  File size: 61

Encryption not required
Signature not required

Destination files sent to recipient address(es) jim.ford@coviantsoftware.com
 from sender address jim@muirford.com
    None

Primary archiving skipped

Additional archiving skipped

No audit record written

Server Address: CoviantLaptop/10.0.0.145
Server Startup Time: March 14, 2016 12:19:17 PM
Diplomat Version: 6.1.2

    </summary>
</diplomatApiResponse>
```

5.4 Log Buffer Request

5.4.1 200 Response Code

```
<diplomatApiResponse responseType="logBufferResponse">
    <statusCode>200</statusCode>
    <bufferNumber>0</bufferNumber>
    <lineNumber>0</lineNumber>
    <line>&amp;gt;Informational      February 27, 2016 5:45:03 PM EST</line>
    <line>Transaction &amp;quot;inftpsold&amp;quot;: Scheduler cancelled.</line>
    ...
    <line>&amp;gt;Informational      March 14, 2016 12:11:24 PM EDT</line>
    <line>Transaction &amp;quot;inftpsold&amp;quot;: Scheduler cancelled.</line>
</diplomatApiResponse>
```

5.5 Log View Request

5.5.1 200 Response Code

```
<diplomatApiResponse responseType="logViewResponse">
    <statusCode>200</statusCode>
    <id>log4276764123321746783.tmp</id>
    <numLines>60</numLines>
    <bufferSize>1000</bufferSize>
</diplomatApiResponse>
```

5.6 Transaction Statuses Request

5.6.1 200 Response Code

```
<diplomatApiResponse responseType="transactionStatusesResponse">
    <statusCode>200</statusCode>
    <monitorTime>1457973007219</monitorTime>
    <dbUpdating>false</dbUpdating>
    <transactionStatusSummary row="0" column="0" value="2"/>
    <transactionStatusSummary row="0" column="1" value="3"/>
    <transactionStatusSummary row="1" column="0" value="1"/>
    <transactionStatusSummary row="1" column="1" value="0"/>
    <transactionStatusSummary row="2" column="0" value="0"/>
    <transactionStatusSummary row="2" column="1" value="0"/>
    <transactionStatusSummary row="3" column="0" value="0"/>
    <transactionStatusSummary row="3" column="1" value="0"/>
    <transactionStatus>
        <monitorTime>1457973007220</monitorTime>
        <transactionId>0000000000000005</transactionId>
        <transactionName>emailtest</transactionName>
        <transactionType>Outbound</transactionType>
        <transactionState>Not Scheduled</transactionState>
        <jobHistory>
            <sequence>0000000000000080</sequence>
            <transactionName>emailtest</transactionName>
            <startTime>1450368571726</startTime>
            <endTime>1450368590373</endTime>
            <status>Complete</status>
            <completionStatus>Successful</completionStatus>
            <filesFound>2</filesFound>
            <filesSuccessful>2</filesSuccessful>
            <filesFailed>0</filesFailed>
            <execAttempt>1</execAttempt>
            <completion>1.0</completion>
            <nextScheduled>1450368627739</nextScheduled>
            <fileHistory>
                <sequence>0000000000000080</sequence>
                <source>file1.txt</source>
                <destination>file1.txt</destination>
                <startTime>1450368571828</startTime>
                <endTime>1450368579734</endTime>
                <status>Complete</status>
```

```
                                        <completionStatus>Successful</completionStatus>
                                        <size>7</size>
                                        <numRetries>0</numRetries>
                                </fileHistory>
                                <fileHistory>
                                        <sequence>0000000000000080</sequence>
                                        <source>file2.txt</source>
                                        <destination>file2.txt</destination>
                                        <startTime>1450368579744</startTime>
                                        <endTime>1450368590340</endTime>
                                        <status>Complete</status>
                                        <completionStatus>Successful</completionStatus>
                                        <size>7</size>
                                        <numRetries>0</numRetries>
                                </fileHistory>
                        </jobHistory>
                </transactionStatus>
                <transactionStatus>
                        <monitorTime>1457973007220</monitorTime>
                        <transactionId>0000000000000004</transactionId>
                        <transactionName>second</transactionName>
                        <transactionType>Outbound</transactionType>
                        <transactionState>3rd Party Scheduler</transactionState>
                        <jobHistory>
                                <sequence>0000000000000091</sequence>
                                <transactionName>second</transactionName>
                                <startTime>1457972461439</startTime>
                                <endTime>1457972474232</endTime>
                                <status>Complete</status>
                                <completionStatus>Successful</completionStatus>
                                <filesFound>1</filesFound>
                                <filesSuccessful>1</filesSuccessful>
                                <filesFailed>0</filesFailed>
                                <execAttempt>1</execAttempt>
                                <completion>1.0</completion>
                                <fileHistory>
                                        <sequence>0000000000000091</sequence>
                                        <source>FilesComplete</source>
                                        <destination>FilesComplete</destination>
                                        <startTime>1457972463039</startTime>
                                        <endTime>1457972474171</endTime>
                                        <status>Complete</status>
                                        <completionStatus>Successful</completionStatus>
                                        <size>61</size>
                                        <numRetries>0</numRetries>
                                </fileHistory>
                        </jobHistory>
                </transactionStatus>
        </diplomatApiResponse>
```