Tektronix communications

ISA THIRD PARTY API REFERENCE

VER. 13.2 - 03/26/2014

The ISA API provides developers access to the rich functionality of ISA in a concise, programmer friendly interface. It utilizes straight forward web services which ensure simplicity and platform independence, so appropriate implementation requires only a clear reference to its syntax - which is the purpose of this document.

Tektronix communications

TABLE OF CONTENTS

INTRODUCTION

CONVENTIONS

Several conventions are used throughout this document to aid the reader and for the sake of brevity.

Base URL

All URLs referenced in the documentation have the following base URL:

`https://isa_server:<port>/isaclient`

and will be referenced as {baseURL} throughout the document.

API Mappings

API mappings are detailed by listing required syntax and providing example HTTP requests and responses. Concise versions are listed in the "Quick Reference" section of the appendix.

Example API Mapping:

Note: header, path, and request parameters are defined in the appendix.

Method Request Parameter Table	
Mapping	{baseURL}/PATH/{pathVar1}{pathVar2}
Method	POST|GET|PUT|DELETE
Request headers (required)	requiredHeader1
Request headers (optional)	optionalHeader1 optionalHeader2
Path variables (required)	pathVar1
Path variables (optional)	pathVar2
Request Parameters (required)	requiredRequestParameter1 requiredRequestParameter2
Request Parameters (optional)	optionalRequestParameter1 optionalRequestParameter2 optionalRequestParameter3

Example method request:

Notes:
- highlighted values indicate required parameters
- body parameters are displayed in plain text for readability, but must be UTF-8 encoded
- all request headers must contain a valid security token.

```
POST {baseURL}/PATH/{pathVar1}/{pathVar2}

securitytoken = asdskl39alkdjo2flks;


requiredHeader1: value
optionalHeader1: value
optionalHeader2: value

requiredRequestParameter1=value
requiredRequestParameter2=value
optionalRequestParameter1=value
optionalRequestParameter2=value
optionalRequestParameter3=value
```

Method Request Response Table	
Content-Type	text/xml;charset=UTF-8
Returned element	Root xml element or other designation

Example response body:

Notes:
-HTTP response header omitted unless significant
-[…] indicates condensed output for brevity

```
<?xml version="1.0" encoding="UTF-8"?>
<usefulData xmlns="http://www.tektronix.com/…">
  […]
 </usefulData >
```

Error Conditions

All possible error conditions cannot be anticipated, therefore they cannot be documented. For interesting cases, the particular error response will be detailed in the API mapping. Otherwise, general HTTP error response codes will be returned.

1. SECURE ACCESS

1.1 HTTPS

The ISA API is accessed through HTTPS. To ensure data privacy, unencrypted HTTP is not supported.

1.2 LICENSING

Third party API functionality requires a server containing a valid license and user with the appropriate user role.

1.3 USER AUTHENTICATION

All requests submitted to the ISA API must include a valid security token in the HTTP header to confirm the user's identity. Tokens are granted at login and are invalidated at logout or session timeout. To prevent a session timeout, a recurring heartbeat request must be issued.

1.3.1 Login

A user is logged in by submitting a request to the API as follows:

Login Request	
Mapping	{baseURL}/login
Method	POST
Request parameters (required)	password username

Example request:

```
POST {baseURL}/login

username: admin
password: changeMe
```

Login Response	
Content-Type	text/html;charset=ISO-8859-1
Returned element	login_result

Example response body 1: valid credentials

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<login_result>
    <securitytoken>9Yt6xwrR2hITAdFQxq5WgVwat4IwP90n</securitytoken>
    <userroles>ROLE_3RD_PARTY_API,ROLE_ADMIN</userroles>
    <projectversion>7.12.2-SNAPSHOT</projectversion>
    <oamServerHost></oamServerHost>
    <isOamAvailable>false</isOamAvailable>
    <idleTimeOut></idleTimeOut>
    <isIdleTimeOut></isIdleTimeOut>
    <detectDuplicatePDUs></detectDuplicatePDUs>
    <enableDisplayFilter>true</enableDisplayFilter>
    <messageRetrievalTimeout></messageRetrievalTimeout>
</login_result>
```

Example response body 2: server at maximum capacity for third-party API requests

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<login_result>
    <securitytoken> Max Sessions Reached</securitytoken>
    <userroles>ROLE_3RD_PARTY_API,ROLE_ADMIN</userroles>
    <projectversion>7.12.2-SNAPSHOT</projectversion>
    <oamServerHost></oamServerHost>
    <isOamAvailable>false</isOamAvailable>
    <idleTimeOut></idleTimeOut>
    <isIdleTimeOut></isIdleTimeOut>
    <detectDuplicatePDUs></detectDuplicatePDUs>
    <enableDisplayFilter>true</enableDisplayFilter>
    <messageRetrievalTimeout></messageRetrievalTimeout>
</login_result>
```

Example response body 3: invalid credentials / other failure

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<login_result>
    <securitytoken></securitytoken>
    <userroles></userroles>
    <projectversion></projectversion>
    <oamServerHost></oamServerHost>
    <isOamAvailable>false</isOamAvailable>
    <idleTimeOut></idleTimeOut>
    <isIdleTimeOut></isIdleTimeOut>
    <detectDuplicatePDUs></detectDuplicatePDUs>
    <enableDisplayFilter>true</enableDisplayFilter>
    <messageRetrievalTimeout></messageRetrievalTimeout>
</login_result>
```

1.3.2 Logout

A user is logged out by submitting a request to the API as follows:

Logout Request	
Mapping	{baseURL}/logout
Method	POST
Request headers (required)	securitytoken

Example request:

```
POST {baseURL}/logout
securitytoken: spzJ1GqclCSPXXrlqCsjoyy2J5h7DPg4
```

Logout Response	
Content-Type	text/html;charset=ISO-8859-1
Returned values	success message

Example response body:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"...>
    <html>
        <head>
            <title>Logout succeeded</title>
        </head>
        <body>
            <form>
            You have been successfully logged out securitytoken=spzJ1GqclCSPXXrlqCsjoyy2J5h7DPg4
            </form>
        </body>
    </html>
```

1.3.3 Heartbeat

A session is “kept alive” by submitting a request to the API as follows:

Heartbeat Request	
Mapping	{baseURL}/heartbeat
Method	POST
Request headers (required)	securitytoken

Example request:

```
POST {baseURL}/heartbeat
securitytoken: spzJ1GqclCSPXXrlqCsjoyy2J5h7DPg4
```

Heartbeat Response	
Content-Type	text/html;charset=ISO-8859-1
Returned values	none

Example response body:

A successful body response has no payload.

2. API MAPPINGS

2.1 GET APPLICATION INFORMATION

Provides access to all of GeoApplications that have metadata populated for them.

Application Info Request	
Mapping	{baseURL}/appinfo/{pluginType}
Method	GET
Path variable (required)	pluginType (spi only)

Example request:

```
GET {baseURL}/appinfo/spi
```

Application Info Response	
Content-Type	text
Returned values	application info

Example response body:

```
#DGT_255.142=Identifier
#DGT_5.-1=DPC
#DGT_255.141=HashUserName
#DGT_11.21=GTTCalledNumber
#TRG_226.1=IpPlusPort
#DGT_11.20=GTTCallingNumber
#DGT_231.64=SSIPAddr
#DGT_231.63=GwIpAddr
#DGT_253.14=MSISDN
#DGT_253.13=IMSI
#TRG_238.14=SRNCToTRNCContainer
 […]
```

2.2 CAPTURE

A capture is a set of call records returned by a user defined search expression. The API provides rich access to create, manage, and mine capture sessions.

2.2.1 Get Captures

Get a list of active capture sessions for a particular user.

Get Captures Request	
Mapping	{baseURL}/capture/captureSession
Method	GET
Request headers (required)	username
Request headers (optional)	userroles

Example request:

```
GET{baseURL}/capture/captureSession

username: admin
```

Get Captures Response	
Content-Type	text/xml;charset=UTF-8
Returned element	captureSessionInfo

Example response body:

```
<?xml version="1.0" encoding='UTF-8'>
<captureSession:captureSessionInfos
   <captureSessionInfo>
        <isa:captureId>isa-server:8080/isa;1</isa:captureId>
        <username>admin</username>
        <name>isa-server:8080/isa;1</name>
        <startTime>1346251920000</startTime>
        <endTime>1346357340000</endTime>
        <type>ATTENDED</type>
        <totalRecords>7</totalRecords>
        <mediaRecords>0</mediaRecords>
        <probeElements>DALLAS_PROBE_1</probeElements>
    </captureSessionInfo>
</captureSession:captureSessionInfos>
```

2.2.2 Start Capture

Start a new capture.

Start Capture Request	
Mapping	{baseURL}/capture
Method	POST
Request headers (optional)	username userroles
Request Parameters (required)	monitoredObjects monitoredObjectType
Request Parameters (optional)	endTime flowApplicationFilterExpression flowMessageInspection flowProtocolFilterExpression g10ProtocolFilterExpression genericFilterExpression gsoftProtocolFilterExpression isPersistence keepMpcTaskOnComplete mediaCapture mediaTypeFilterExpression monitoredObjectsDisplayName pluginType pointcodeFilterExpression searchTarget spiProtocolFilterExpression startTime timeTarget topologyType workflowName

Example request:

```
POST {baseURL}/capture

startTime=2012-10-21T06:52:00.000
monitoredObjects=DALLAS_PROBE_1
monitoredObjectType=Probe
workflowName=Session Trace
```

Start Capture Response	
Content-Type	text/xml;charset=UTF-8
Returned element	captureId

Example response body:

```
<?xml version="1.0" encoding="UTF-8"?>
<startCapture>
  <isa:captureId>isa-server:8080/isa;1</isa:captureId>
</startCapture>
```

2.2.3 Stop Capture

Stop a capture.

Stop Capture Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/capture (When nginx is turned off)	{baseURL}/InstanceId/capture (When nginx is turned on)
Method	PUT	
Request headers (required)	captureId	

Example request:

```
PUT {baseURL}/capture
captureId: isa-server:8080/isa;1
```

Stop Capture Response	
Content-Type	text/xml;charset=UTF-8
Returned element	captureId

Example response body:

```
<?xml version="1.0" encoding="UTF-8"?>
<stopCapture>
  <isa:captureId>isa-server:8080/isa;1</isa:captureId>
</stopCapture>
```

2.2.4 Close Capture

Close a running capture.

Close Capture Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/capture	{baseURL}/InstanceId/capture
Method	DELETE	
Request headers (required)	captureId	

Example request:

```
DELETE {baseURL}/capture
captureId: isa-server:8080/isa;1
```

Close Capture Response	
Content-Type	text/xml;charset=UTF-8
Returned element	captureId

Example response body:

```
<?xml version="1.0" encoding="UTF-8"?>
<closeCapture>
  <isa:captureId>isa-server:8080/isa;1</isa:captureId>
</closeCapture>
```

2.2.5 Resume Capture

Resume a paused capture.

Resume Capture Request		
	When nginx is turned OFF	**When nginx is turned ON (if different from, when turned OFF)**
Mapping	{baseURL}/capture/resume	{baseURL}/InstanceId/capture/resume
Method	PUT	
Request headers (required)	captureId	

Example request:

```
PUT {baseURL}/capture/resume
captureId: isa-server:8080/isa;1
```

Resume Capture Response	
Content-Type	text/xml;charset=UTF-8
Returned element	captureId

Example response body:

```
<?xml version="1.0" encoding="UTF-8">
<isa:resumeCapture>
    <isa:captureId>isa-server:8080/isa;1</isa:captureId>
<isa:resumeCapture>
```

2.2.6 Attach Capture

Resume a detached (unattended) capture.

Attach Capture Request		
	When nginx is turned OFF	**When nginx is turned ON (if different from, when turned OFF)**
Mapping	{baseURL}/capture/attach	{baseURL}/InstanceId/capture/attach
Method	PUT	
Request headers (required)	captureId	

Example request:

```
PUT {baseURL}/capture/attach
captureId: isa-server:8080/isa;1
```

Attach Capture Response	
Content-Type	text/xml;charset=UTF-8
Returned element	none

Example response body:

A successful body response has no payload.

2.2.7 Detach Session

Initiate a resumable detached (unattended) session.

Detach Session Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/capture/detach	{baseURL}/InstanceId/capture/detach
Method	POST	
Request headers (required)	captureId username	
Request parameters (required)	duration name	

Example request:

```
POST {baseURL}/capture/detach
captureId: isa-server:8080/isa;1
username: user
name=sessionName
duration=60
```

Detach Session Response	
Content-Type	text/xml;charset=UTF-8
Returned element	none

Example response body:

A successful body response has no payload.

2.2.8 Terminate Sessions

Terminate a session(s) for a given captureId(s).

Terminate Sessions Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/capture/terminate	{baseURL}/InstanceId/capture/terminate
Method	DELETE	
Request headers (required)	username	
Request Parameters (required)	captureId	

Example request:

```
DELETE {baseURL}/capture/terminate
username: user
captureId= isa-server:8080/isa;1, isa-server:8080/isa;2
```

Terminate Sessions Response	
Content-Type/Encoding	text/xml;gzip
Returned element	none

Example response body:

```
A successful body response has no payload.
```

2.2.9 Touch Capture

Prevent VIP captures from being closed by the aging handler.

Touch Capture Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/capture/touch	{baseURL}/InstanceId/capture/touch
Method	PUT	
Request headers (required)	captureId	

Example request:

```
PUT {baseURL}/capture/touch
captureId: isa-server:8080/isa;1
```

Touch Capture Response	
Content-Type	text/xml;charset=UTF-8
Returned element	success message

Example response body:

```
Success
```

2.2.10 Set Post Capture Filter Session

Set a capture filter after a capture has been started.

Post Capture Filter Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/capture/post	{baseURL}/InstanceId/capture/post
Method	POST	
Request headers (required)	captureId	
Request parameters (optional)	filterExpression, timezone	

Example request:

```
POST {baseURL}/capture/post
captureId: isa-server:8080/isa;1
filterExpression: ('IPADDR' = '10.0.0.1')
timezone: America/Chicago
```

Set Post Capture Filter Response	
Content-Type	text/xml;charset=UTF-8/gzip
Returned element	postCapture

Example response body:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<postCapture>
    <isa:captureId>camila-vm1:11300/isa;11</isa:captureId>
<postCapture>
```

2.2.11 Detach Enable

Checks if capture session detachment is enabled or not.

Detach Enable Request	
Mapping	{baseURL}/capture/detachEnable
Method	POST
Request headers (required)	none
Request parameters (optional)	none

Example request:

```
POST {baseURL}/capture/detachEnable
```

Detach Enable Response	
Content-Type	text/html;charset=UTF-8
Returned element	none

Example response body:

```
Returns Status code 200 if detach is enabled and Status code 403 if not.
```

2.2.12 Get Chunk Records

Get matching records in configurable quantities per RESTful poll.

Get Chunk Records Request		
	When nginx is turned OFF	**When nginx is turned ON (if different from, when turned OFF)**
Mapping	{baseURL}/capture/records/{chunkSize}	{baseURL}/InstanceId/capture/records/{chunkSize}
Method	GET	
Request headers (required)	captureId	
Path variable (required)	chunkSize	

Example request:

```
GET {baseURL}/capture/records/3
captureId: isa-server:8080/isa;1
```

Get Chunk Records Response	
Content-Type/Encoding	text/xml;charset=UTF-8/gzip
Returned element	records

Example response body:

```
<?xml version="1.0" encoding="UTF-8">
<records:records>
    <isa:captureId>isa-server:8080/isa;1</isa:captureId>
    <recordTransaction action="ADD">
        <record>
            <isa:recordId>
                <pluginType>mpc</pluginType>
                <pluginId>863175514</pluginId>
            </isa:recordId>
            <mpcState>QUICK</mpcState>
            <timeRangeState>FULL</timeRangeState>
            <entity name="SessionId" type="ID_STRING">
                <instance>
                    <attribute>
                        <name>recordId</name>
                        <value>mpc:863175514</value>
                    </attribute>
                </instance>
            </entity>
            [...]
        </record>
    </recordTransaction>
    <recordTransaction action="ADD">
        <record>
            <isa:recordId>
                <pluginType>mpc</pluginType>
                <pluginId>524764653</pluginId>
            </isa:recordId>
            <mpcState>QUICK</mpcState>
            <timeRangeState>PARTIAL</timeRangeState>
            <entity name="SessionId" type="ID_STRING">
                <instance>
                    <attribute>
                        <name>recordId</name>
                        <value>mpc:524764653</value>
                    </attribute>
                </instance>
            </entity>
                 [...]
        </record>
    </recordTransaction>
    <recordTransaction action="UPDATE">
        <record>
            <isa:recordId>
                <pluginType>mpc</pluginType>
                <pluginId>481425680</pluginId>
            </isa:recordId>
            <mpcState>QUICK</mpcState>
            <timeRangeState>FULL</timeRangeState>
            <entity name="SessionId" type="ID_STRING">
                <instance>
                    <attribute>
                        <name>recordId</name>
                        <value>mpc:481425680</value>
                    </attribute>
                </instance>
            </entity>
            [...]
        </record>
    </recordTransaction>
</records:records>
```

2.2.13 Get All Records

Get all matching records.

Get All Records Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/capture/records	{baseURL}/InstanceId/capture/records
Method	GET	
Request headers (required)	captureId	

Example request:

```
GET {baseURL}/capture/records
captureId: isa-server:8080/isa;1
```

Get All Records Response	
Content-Type/Encoding	text/xml;charset=UTF-8/gzip
Returned element	records

Example response body:

```
<?xml version="1.0" encoding="UTF-8”?>
<records:records>
    <isa:captureId>isa-server:8080/isa;1</isa:captureId>
    <recordTransaction action="ADD">
        <record>
            <isa:recordId>
                <pluginType>mpc</pluginType>
                <pluginId>863175514</pluginId>
            </isa:recordId>
            <mpcState>QUICK</mpcState>
            <timeRangeState>FULL</timeRangeState>
            <entity name="SessionId" type="ID_STRING">
                <instance>
                    <attribute>
                        <name>recordId</name>
                        <value>mpc:863175514</value>
                    </attribute>
                </instance>
            </entity>
            [...]
        </record>
    </recordTransaction>
    <recordTransaction action="ADD">
        <record>
            <isa:recordId>
                <pluginType>mpc</pluginType>
                <pluginId>524764653</pluginId>
            </isa:recordId>
            <mpcState>QUICK</mpcState>
            <timeRangeState>PARTIAL</timeRangeState>
            <entity name="SessionId" type="ID_STRING">
                <instance>
                    <attribute>
                        <name>recordId</name>
                        <value>mpc:524764653</value>
                    </attribute>
                </instance>
            </entity>
                 [...]
        </record>
    </recordTransaction>
    <recordTransaction action="UPDATE">
        <record>
            <isa:recordId>
                <pluginType>mpc</pluginType>
                <pluginId>481425680</pluginId>
            </isa:recordId>
            <mpcState>QUICK</mpcState>
            <timeRangeState>FULL</timeRangeState>
            <entity name="SessionId" type="ID_STRING">
                <instance>
                    <attribute>
                        <name>recordId</name>
                        <value>mpc:481425680</value>
                    </attribute>
                </instance>
            </entity>
            [...]
        </record>
    </recordTransaction>
    [...]
</records:records>
```

2.2.14 Monitored Elements

Get monitored elements.

Get Monitored Elements Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/capture/monitoredElements	{baseURL}/InstanceId/capture/monitoredElements
Method	GET	
Request headers (required)	captureId	

Example request:

```
GET {baseURL}/monitoredElements
captureId: isa-server:8080/isa;1
```

Get Monitored Elements Response	
Content-Type	text/xml;charset=UTF-8
Returned element	monitoredElements

Example response body:

```
<?xml version="1.0" encoding="UTF-8"?>
<monitoredElements:monitoredElements>
    <monitoredObjects>
        <monitoredObject>
            <displayName>DALLAS_PROBE_1</displayName>
            <groupNames/>
            <monitoredIds>
                <monitoredId>
                    <pluginId>Probe:4097</pluginId>
                    <pluginType>g10</pluginType>
                </monitoredId>
            </monitoredIds>
            <networkType>Probe</networkType>
            <providers/>
        </monitoredObject>
    </monitoredObjects>
</monitoredElements:monitoredElements>
```

2.3 EVENTS

Events provide feedback on capture status.

2.3.1 Get Events without Time

Get events irrespective of time.

Get Events without Time Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/event	{baseURL}/InstanceId/event
Method	GET	
Request headers (required)	captureId	

Example request:

```
GET {baseURL}/event
captureId: isa-server:8080/isa;1
```

Get Events without Time Response	
Content-Type/Encoding	text/xml;charset=UTF-8/gzip
Returned element	events

Example response body:

```
<?xml version="1.0" encoding="UTF-8"?>
<events:events>
    <event>
        <isa:captureId>isa-server:8080/isa;1</isa:captureId>
        <type>server.status.probeQueryState</type>
        <attributes>
            <attribute>
                <key>timestamp</key>
                <value>08/31/2012 16:22:09.353000000 +0000</value>
            </attribute>
        </attributes>
        <arguments>
            <argument>DALLAS_PROBE_1</argument>
            <argument>PAUSED</argument>
        </arguments>
        <flags/>
    </event>
    <event>
        <isa:captureId> isa-server:8080/isa;1</isa:captureId>
        <type>server.status.probeQueryState</type>
        <attributes>
            <attribute>
                <key>timestamp</key>
                <value>08/31/2012 16:22:09.553000000 +0000</value>
            </attribute>
        </attributes>
        <arguments>
            <argument>DALLAS_PROBE_1</argument>
            <argument>COMPLETE</argument>
        </arguments>
        <flags/>
    </event>
```

```
    <event>
        <isa:captureId>isa-server:8080/isa;1</isa:captureId>
        <type>server.status.captureProgress</type>
        <attributes>
            <attribute>
                <key>timestamp</key>
                <value>08/31/2012 16:22:09.561000000 +0000</value>
            </attribute>
            <attribute>
                <key>PLUGIN_TYPE</key>
                <value>G10</value>
            </attribute>
        </attributes>
        <arguments>
            <argument>100</argument>
        </arguments>
        <flags/>
    </event>
    <event>
        <isa:captureId>isa-server:8080/isa;1</isa:captureId>
        <type>server.status.historical.complete</type>
        <attributes>
            <attribute>
                <key>timestamp</key>
                <value>08/31/2012 16:22:09.563000000 +0000</value>
            </attribute>
        </attributes>
        <arguments>
            <argument>DALLAS_PROBE_1</argument>
            <argument>1</argument>
            <argument>1</argument>
        </arguments>
        <flags/>
    </event>
    <event>
        <isa:captureId>isa-server:8080/isa;1</isa:captureId>
        <type>server.status.recordLimitProgress</type>
        <attributes>
            <attribute>
                <key>timestamp</key>
                <value>08/31/2012 16:22:14.678000000 +0000</value>
            </attribute>
        </attributes>
        <arguments>
            <argument>9</argument>
            <argument>1000</argument>
        </arguments>
        <flags/>
    </event>
</events:events>
```

2.3.2 Get Events with Time

Get events since a specific last event time.

Get Events with Time Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/ event/{lastEventTime}	{baseURL}/InstanceId/event/{lastEventTime}
Method	GET	
Request headers (required)	captureId	
Path variable (required)	lastEventTime	

Example request:

```
GET {baseURL}/event/1346266281696
captureId: isa-server:8080/isa;1
```

Get Events with Time Response	
Content-Type/Encoding	text/xml;charset=UTF-8/gzip
Returned element	events

Example response body:

```
<?xml version="1.0" encoding="UTF-8”?>
<events:events>
    <event>
        <isa:captureId>isa-server:8080/isa;1</isa:captureId>
        <type>server.status.probeQueryState</type>
        <attributes>
            <attribute>
                <key>timestamp</key>
                <value>08/31/2012 16:22:09.353000000 +0000</value>
            </attribute>
        </attributes>
        <arguments>
            <argument>DALLAS_PROBE_1</argument>
            <argument>PAUSED</argument>
        </arguments>
        <flags/>
    </event>
    <event>
        <isa:captureId> isa-server:8080/isa;1</isa:captureId>
        <type>server.status.probeQueryState</type>
        <attributes>
            <attribute>
                <key>timestamp</key>
                <value>08/31/2012 16:22:09.553000000 +0000</value>
            </attribute>
        </attributes>
        <arguments>
            <argument>DALLAS_PROBE_1</argument>
            <argument>COMPLETE</argument>
        </arguments>
        <flags/>
    </event>
    <event>
        <isa:captureId>isa-server:8080/isa;1</isa:captureId>
        <type>server.status.captureProgress</type>
        <attributes>
```

```
            <attribute>
                <key>timestamp</key>
                <value>08/31/2012 16:22:09.561000000 +0000</value>
            </attribute>
            <attribute>
                <key>PLUGIN_TYPE</key>
                <value>G10</value>
            </attribute>
        </attributes>
        <arguments>
            <argument>100</argument>
        </arguments>
        <flags/>
    </event>
    <event>
        <isa:captureId>isa-server:8080/isa;1</isa:captureId>
        <type>server.status.historical.complete</type>
        <attributes>
            <attribute>
                <key>timestamp</key>
                <value>08/31/2012 16:22:09.563000000 +0000</value>
            </attribute>
        </attributes>
        <arguments>
            <argument>DALLAS_PROBE_1</argument>
            <argument>1</argument>
            <argument>1</argument>
        </arguments>
        <flags/>
    </event>
    <event>
        <isa:captureId>isa-server:8080/isa;1</isa:captureId>
        <type>server.status.recordLimitProgress</type>
        <attributes>
            <attribute>
                <key>timestamp</key>
                <value>08/31/2012 16:22:14.678000000 +0000</value>
            </attribute>
        </attributes>
        <arguments>
            <argument>9</argument>
            <argument>1000</argument>
        </arguments>
        <flags/>
    </event>
</events:events>
```

2.4 EXPORT

Export provides capability of viewing or processing ISA generated data in a third-party application.

2.4.1 Start Export

Export a session.

Start Export Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/export/doExport/{exportType}/{recordIdsCsv}	{baseURL}/InstanceId/export/doExport/{exportType}/{recordIdsCsv}
Method	POST	
Request headers (required)	captureId	
Path variable (required)	exportType recordIdsCsv	
Request Parameters (required)	requestId content (for CSV)	
Request Parameters (optional) (PCAP)	allRecords currentLadderDiagramRecord doFullMpc fileSize fileSizeUnits groupByRecord includedMessages merge messageEndTime messageStartTime split splitByType splitFileSize splitFileSizeUnits withUserPlanePDUs	
Request Parameters (optional) (CSV)	gmtOffset includedMessages messageTableColumns recordTableColumns	

Request Parameters (optional) (HTML)	absoluteTime endPointType gmtOffset groupedNodeTypes header includedMessages ladder messageDetails messageTable messageTableColumns nodes overriddenNodes pinnedNodes recordTable recordTableColumns	
Request Parameters (optional) (ISA)	allGroupNodeTypes allRecords areMessagesFiltered comment currentLadderDiagramRecord doFullMpc doMessageRetrieval groupedNodeTypes includedMessages messageColumnSetName messageEndTime messageStartTime messageTableColumns nodeOrdering nodeOrderingEnabled nodeTypes overriddenNodeTypes pinnedNodes recordColumnSetName recordTableColumns	

Example HTTP export request:

```
POST {baseURL}/export/doExport/HTML/mpc:12326548
captureId: isa-server:8080/isa;1
requestId = 1001
recordTable = true
ladder=true
messageTable=true
endPointType=IP Layer1
nodes=MME/108.188.1.1|SGW/100.1.1.1
recordTableColumns=SessionId$-$IMEISV$-$TransportProtocols$-$ECGI$-$ApplicationId
messageTableColumns=Start Time$-$End Time
```

Start HTTP Export Response	
Content-Type	text/xml;charset=UTF-8
Returned element	html

Example HTTP export response body:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN">
<html xmlns="http://www.w3.org/1999/xhtml">
[...]
</html>
```

Example HTML Report:

Tektronix®
Communications

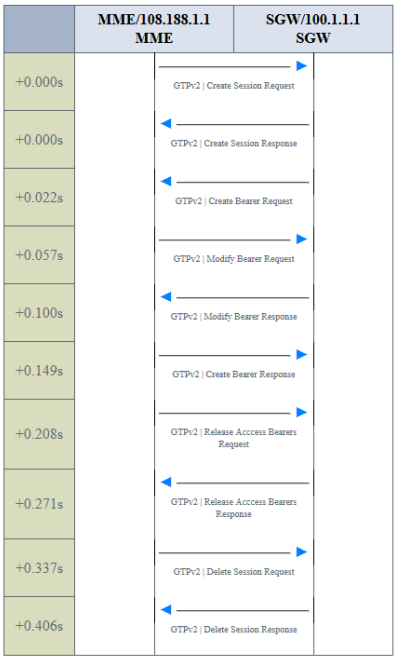
Session Summary Table

Session Ladder | PDU Table

SessionID	IMEISV	TransportProtocols	ECGI	ApplicationID
mpc:2135661500	990000450910XXXX	UDP	3114800081101, 3114800081103	GTPv2

Session Ladder Diagram

Session Table | PDU Table



Session PDU Table

Session Table | Session Ladder

Ladder	Start Time	End Time
⇄	10/22/2012 23:41:08.875051700 +0000	10/22/2012 23:41:08.875051700 +0000
⇄	10/22/2012 23:41:08.875143280 +0000	10/22/2012 23:41:08.875143280 +0000
⇄	10/22/2012 23:41:08.897485580 +0000	10/22/2012 23:41:08.897485580 +0000
⇄	10/22/2012 23:41:08.932490880 +0000	10/22/2012 23:41:08.932490880 +0000
⇄	10/22/2012 23:41:08.975495040 +0000	10/22/2012 23:41:08.975495040 +0000
⇄	10/22/2012 23:41:09.024477740 +0000	10/22/2012 23:41:09.024477740 +0000
⇄	10/22/2012 23:41:09.083507680 +0000	10/22/2012 23:41:09.083507680 +0000
⇄	10/22/2012 23:41:09.146490380 +0000	10/22/2012 23:41:09.146490380 +0000
⇄	10/22/2012 23:41:09.212499240 +0000	10/22/2012 23:41:09.212499240 +0000
⇄	10/22/2012 23:41:09.281498380 +0000	10/22/2012 23:41:09.281498380 +0000

Example CSV export request:

```
POST {baseURL}/export/doExport/CSV/mpc:12326548
captureId: isa-server:8080/isa;1
requestId = 1001
content=messages
```

Start CSV Export Response	
Content-Type	text/xml;charset=UTF-8
Returned element	csv

Example CSV export response body:

```
"StartDate","StartTime","StartDateMillis","StartTimeMillis","StartTimeNanos","EndDate","EndTime"
,"EndDateMillis","EndTimeMillis","EndTimeNanos"
"2012-10-22","23:41:08.875051700",1350864000000,1350949268000,875051700,"2012-10-
22","23:41:08.875051700",1350864000000,1350949268000,875051700
22","23:41:09.212499240",1350864000000,1350949269000,212499240
"2012-10-22","23:41:09.281498380",1350864000000,1350949269000,281498380,"2012-10-
22","23:41:09.281498380",1350864000000,1350949269000,281498380
```

2.4.2 Export Now

Force the return of a file from an export already in progress with only the records/messages currently in server datastore.

Export Now Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/export/exportNow/{requestId}	{baseURL}/InstanceId/export/exportNow/{requestId}
Method	PUT	
Request headers (required)	captureId	
Path variable (required)	requestId	

Example request:

```
PUT {baseURL}/exportNow/1001
captureId: isa-server:8080/isa;1
messageId = 1001
```

Export Now Response	
Content-Type	text/xml;charset=UTF-8
Returned element	html

Example response body:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN">
<html xmlns="http://www.w3.org/1999/xhtml">
[...]
</html>
```

2.4.3 Cancel Now

Cancel an active export.

Cancel Now Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/ export/cancelNow/{requestId}	{baseURL}/InstanceId/ export/cancelNow/{requestId}
Method	PUT	
Request headers (required)	captureId	
Path variable (required)	requestId	
Request parameters (optional)	doExportToRepository exportTime fileName requestId username	

Example request:

```
PUT {baseURL}/cancelNow/1001
captureId: isa-server:8080/isa;1
```

Cancel Now Response	
Content-Type	text/xml;charset=UTF-8
Returned element	none

Example response body:

A successful body response has no payload.

2.4.4 Get Plist Properties

Get Plist properties.

Get Plist Properties Request	
Mapping	{baseURL}/export/getPlistProperties
Method	GET

Example request:

```
GET {baseURL}/export/getPlistProperties
```

Get Plist Properties Response	
Content-Type	text/xml;charset=UTF-8
Returned element	properties

Example response body:

```
{properties}
```

2.5 FILTERS

Filters refine a search by limiting scope of capture session.

2.5.1 Get Filter Elements

Gets filter elements.

Get Filter Elements Request	
Mapping	{baseURL}/filter/{pluginTypes}
Method	GET
Request headers (optional)	username userroles
Path variables (required)	pluginTypes

Example request:

```
GET {baseURL}/filter/g10
```

Get Filter Elements Response	
Content-Type	text/xml;charset=UTF-8
Returned element	filters

Example response body:

```
<?xml version="1.0" encoding="UTF-8">
<filters>
    <filter>
        <name>ESN</name>
        <group>DIGIT</group>
        <probeType>spi</probeType>
        <operation selected="true">Matches Any</operation>
            <supportedOptions>
            </supportedOptions>
            <values>
            </values>
            <rules>
                <validationRule>hexDigitValidator</validationRule>
            </rules>
    </filter>
    [...]
    <filter>
        <name>IpRange</name>
        <group>DIGIT</group>
        <probeType>g10</probeType>
        <operation selected="true">Matches Any</operation>
            <supportedOptions>
            </supportedOptions>
            <values>
            </values>
            <rules>
                <validationRule>ipRangeValidator</validationRule>
            </rules>
    </filter>
    [...]
    <filter>
        <name>CallingParty</name>
        <group>DIGIT</group>
        <probeType>common</probeType>
        <operation selected="true">Matches Any</operation>
            <supportedOptions>
            </supportedOptions>
            <values>
            </values>
            <rules>
                <validationRule>hexDigitG10Validator</validationRule>
                <validationRule>hexDigitValidator</validationRule>
            </rules>
    </filter
</:filters>
```

2.5.2 Get Applications

Get applications for a plugin type.

Get Applications Request	
Mapping	{baseURL}/filter/application/{pluginType}
Method	GET
Request headers (optional)	none
Path variables (required)	pluginTypes

Example request:

```
GET {baseURL}/filter/application/g10
```

Get Applications Response	
Content-Type	text/xml;charset=UTF-8
Returned element	applications

Example response body:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<applications>
    <application>
        <id>265</id>
        <name>ISAIC</name>
        <protocols>61013</protocols>
        <supportedProbeTypes>g10</supportedProbeTypes>
    </application>
    <application>
        <id>5</id>
        <name>GSM MAP</name>
        <protocols>61026</protocols>
        <supportedProbeTypes>g10</supportedProbeTypes>
    </application>
    <application>
        <id>218</id>
        <name>RTSP</name>
        <protocols>554</protocols>
        <supportedProbeTypes>g10</supportedProbeTypes>
    </application>
    <application>
        <id>224</id>
        <name>DSS1</name>
        <protocols>61004</protocols>
        <supportedProbeTypes>g10</supportedProbeTypes>
    </application>
</applications>
```

2.5.3 Get Advanced Options

Get advanced filter options.

Get Advanced Options Request	
Mapping	{baseURL}/filter/advanced/{pluginType}
Method	GET
Request headers (optional)	userroles
Path variables (required)	pluginTypes

Example request:

```
GET {baseURL}/filter/advanced/g10
```

Get Advanced Options Response	
Content-Type	text/xml;charset=UTF-8
Returned element	advancedOptions

Example response body:

```
<?xml version="1.0" encoding="UTF-8">
<advancedOptions>
    <advancedOption>
        <name>Media Type</name>
        <value>false</value>
        <probeType>g10</probeType>
        <advancedOptionType>MEDIATYPE</advancedOptionType>
    </advancedOption>
    <advancedOption>
        <name>Flow Protocol</name>
        <value>false</value>
        <probeType>g10</probeType>
        <advancedOptionType>FLOWPROTOCOL</advancedOptionType>
    </advancedOption>
    <advancedOption>
        <name>Flow Application</name>
        <value>false</value>
        <probeType>g10</probeType>
        <advancedOptionType>FLOWAPPLICATION</advancedOptionType>
    </advancedOption>
</advancedOptions>
```

2.5.4 Get Flow Applications

Get a list of flow applications.

Get Flow Applications Request	
Mapping	{baseURL}/filter/advanced/flowApplications
Method	GET

Example request:

```
GET {baseURL}/filter/advanced/flowApplications
```

Get Flow Applications Response	
Content-Type	text/xml;charset=UTF-8
Returned element	flow applications

Example response body:

```
["Flickr","Hamachi","HTTP Video", "Vonage", "Docstoc", "ApplePNS", "Last.fm", "FileServe",
"LogMein","Hotmail","BitDefender","Wikipedia","Xunlei","SkyDrive","Google Desktop",
"Grooveshark","Twitter","DynGate","Usenet","Paltalk Chat","AD DSROL","MS Online","NetFlix
Video","Craigslist","MultiUpload","AVG","F-Prot","Kaspersky","Skype Auth", "IsoHunt", "Imgur",
"Yelp","Dropbox","Skype","MafiaWars","AD Name Service","MUTE-net","Google Docs", "Steam",
"Tumblr","CNET","Avira","TeamViewer","FreeeTV","Metacafe","Hopster","Privax","MS RFI",
"Yahoo","MSN","MS Exchange","Paltalk IM","Adobe","BitTorrent","Android
C2DM","iMesh","4Shared","Fogbugz","WebEx","Zynga","AD Restore","Picasa","MS MTA","Google
Earth","Filesonic","Manolito","eBay","Classmates","SuperNews","MSDN","YouTube","iCloud","Line2",
"MySpace","Baidu","Google Analytics","Xanga","XML","ApplJuice","BlackBerry","TwitPic","MS
MAPI","MagicJack
[...]
Online","Giganews","MS Store Admin","Facebook"]
```

2.6 FILTER VALIDATION

Filter validation provides a means of confirming correct syntax of potentially complicated filter expressions.

2.6.1 Validate Filters

Validates generic filters.

Validate Filters Request	
Mapping	{baseURL}/filterExpressions/validate
Method	POST
Request Parameters (required)	monitoredObjects
Request Parameters (optional)	monitoredObjectType genericFilterExpression g10ProtocolFilterExpression spiProtocolFilterExpression pointcodeFilterExpression mediaTypeFilterExpression flowProtocols flowApplications

Example request:

```
POST {baseURL}/filterExpressions/validate
monitoredObjects=DALLAS_PROBE
genericFilterExpression=('IPADDR' = '10.0.0.1')
```

Validate Filters Response	
Content-Type	text/xml;charset=UTF-8
Returned element	validation result

Example response body:

OK

2.6.2 Validate Proto Filter

Validates a protocol filter.

Validate Proto Filter Request	
Mapping	{baseURL}/filter/validation/proto
Method	POST
Request Parameters (required)	expression probeType

Example request:

```
POST {baseURL}/filter/validation/proto
expression='GTPv2':('Response Code' = 'GTPv2-C;;4/RAT Changed from 3GPP to Non-3GPP' )
probeType=g10
```

Validate Proto Filters Response	
Content-Type	text/xml;charset=UTF-8
Returned element	validation result

Example response body:

OK

2.6.3 Validate Digit Filter

Validates a digit filter.

Validate Digit Filter Request	
Mapping	{baseURL}/filter/validation/digit
Method	POST
Request Parameters (required)	expression pluginTypes

Example request:

```
POST {baseURL}/filter/validation/digit
expression=('IPADDR' = '10.0.0.1')
pluginTypes=g10
```

Validate Digit Filters Response	
Content-Type	text/xml;charset=UTF-8
Returned element	validation result

Example response body:

OK

2.6.4 Validate Flow Applications

Validates flow applications.

Validate Flow Applications Request	
Mapping	{baseURL}/filter/validation/flowApplications
Method	POST
Request Parameters (required)	flowApplications

Example request:

```
POST {baseURL}/filter/validation/flowApplications
flowApplications: AIM, Vonage
```

Validate Flow Applications Response	
Content-Type	text/xml;charset=UTF-8
Returned element	validation result

Example response body:

OK

2.6.5 Validate Flow Protocols

Validates flow protocols.

Validate Flow Protocols Request	
Mapping	{baseURL}/filter/validation/flowProtocols
Method	POST
Path variables (optional)	flowProtocols

Example request:

```
POST {baseURL}/filter/validation/flowProtocols
flowProtocols=DHCP, DNS
```

Validate Flow Protocols Response	
Content-Type	text/xml;charset=UTF-8
Returned element	validation result

Example response body:

OK

2.7 MESSAGES

A message is, in concept, a basic component of a record. Examples of messages are PDUs and flow summaries. Messages usually represent an individual transaction of some kind between two endpoints. API mappings for messages use the various methods on URL: {baseUrl/capture/{recordId}.

2.7.1 Start Messages

Start messages.

Start Messages Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/capture/{recordId}	{baseURL}/InstanceId/capture/{recordId}
Method	POST	
Request headers (required)	captureId	
Path variables (required)	recordId	

Example request:

```
POST {baseURL}/capture/mpc:1577166801

captureId: isa-server:8080/isa;1
```

Start Messages Response	
Content-Type/Encoding	text/xml;charset=UTF-8/gzip
Returned element	startMessages

Example response body:

```
<?xml version="1.0" encoding="UTF-8"?>
<msg:startMessages>
  <isa:captureId>server:8080/isa;1</isa:captureId>
  <isa:recordId>
    <type>mpc</type>
    <identifier>1577166801</identifier>
  </isa:recordId>
</msg:startMessages>
```

2.7.2 Get Messages

Get messages.

Get Messages Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/capture/{recordId}	{baseURL}/InstanceId/capture/{recordId}
Method	GET	
Request headers (optional)	captureId	
Path variable (required)	recordId	
Request parameter (optional)	Index pageSize	

Example request:

```
GET {baseURL}/capture/mpc:1577166801
captureId: isa-server:8080/isa;1
```

Get Messages Response	
Content-Type/Encoding	text/xml;charset=UTF-8/gzip
Returned element	messages

Example response body:

```
<?xml version="1.0" encoding="UTF-8">
<isa:messages>
    <isa:captureId>isa-server:8080/isa;1</isa:captureId>
    <isa:recordId>
        <pluginType>mpc</pluginType>
        <pluginId>1577166801</pluginId>
    </isa:recordId>
    <totalMessages>2</totalMessages>
    <totalPdus>2</totalPdus>
    <totalFlows>0</totalFlows>
    <totalFlowUpdates>0</totalFlowUpdates>
    <totalTxUpdates>0</totalTxUpdates>
</isa:messages>
```

2.7.3 Stop Messages

Stop Messages.

Stop Messages Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/capture/{recordId}	{baseURL}/InstanceId/capture/{recordId}
Method	DELETE	
Request headers (required)	captureId	
Path variable (required)	recordId	

Example request:

```
DELETE {baseURL}/capture/ mpc:1577166801
captureId: isa-server:8080/isa;1
```

Stop Messages Response	
Content-Type/Encoding	text/xml;charset=UTF-8/gzip
Returned element	stopMessages

Example response body:

```
<?xml version="1.0" encoding="UTF-8">
<msg:stopMessages">
  <isa:captureId>server:port/isa;1</isa:captureId>
  <isa:recordId>
    <type>mpc</type>
    <identifier>1577166801</identifier>
  </isa:recordId>
</msg:stopMessages>
```

2.8 MESSAGE DETAILS

Message details return specific PDU and Flow data.

2.8.1 Start Message Details

Initiate retrieval of message details.

Start Message Details Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/capture/{recordId}/{messageId}	{baseURL}/InstanceId/capture/{recordId}/{messageId}
Method	POST	
Request headers (optional)	captureId	
Path variable (required)	messageId recordId	

Example request:

```
POST {baseURL}/capture/mpc:1577166801/g10:FLOW:4097.17.3.4206064828416.0.0.0
captureId: isa-server:8080/isa;1
```

Start Message Details Response	
Content-Type/Encoding	text/xml;charset=UTF-8/gzip
Returned element	startMessageDetails

Example response body:

```
<?xml version="1.0" encoding="UTF-8">
<isa:startMessageDetails[...]">
    <isa:captureId>localhost:8080/isa;3</ns3:captureId>
    <isa:recordId>
        <pluginType>mpc</pluginType>
        <pluginId>2122219025</pluginId>
    </isa:recordId>
    <messageId>
        <pluginId>FLOW:4097.17.3.4206064828416.0.0.0</pluginId>
        <pluginType>g10</pluginType>
    </messageId>
</isa:startMessageDetails>
```

2.8.2 Stop Message Details

Stop retrieval of message details.

Stop Message Details Request		
	When nginx is turned OFF	**When nginx is turned ON (if different from, when turned OFF)**
Mapping	{baseURL}/capture/{recordId}/{messageId}	{baseURL}/InstanceId/capture/{recordId}/{messageId}
Method	DELETE	
Request headers (optional)	captureId	
Path variable (required)	messageId recordId	
Request Parameter (required)	stopAction	

Example request:

```
DELETE {baseURL}/capture/mpc:1577166801/g10:FLOW:4097.17.3.4206064828416.0.0.0
captureId: isa-server:8080/isa;1

stopAction=STOP
```

Stop Message Details Response	
Content-Type/Encoding	text/xml;charset=UTF-8/gzip
Returned element	stopMessageDetails

Example response body:

```
<?xml version="1.0" encoding="UTF-8">
<isa:stopMessageDetails>
    <isa:captureId>localhost:8080/isa;1</ns3:captureId>
    <isa:recordId>
        <pluginType>mpc</pluginType>
        <pluginId>760301274</pluginId>
    </isa:recordId>
    <messageId>
        <pluginId>FLOW:4097.17.3.4206064828416.0.0.0</pluginId>
        <pluginType>g10</pluginType>
    </messageId>
</isa:stopMessageDetails
```

2.8.3 Get Message Details Status

Get message details status.

Get Message Details Status Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/capture/{recordId}/{messageId}	{baseURL}/InstanceId/capture/{recordId}/{messageId}
Method	PUT	
Request headers (optional)	captureId	
Path variable (required)	messageId recordId	

Example request:

```
PUT {baseURL}/capture/mpc:1577166801/g10:FLOW:4097.17.3.4206064828416.0.0.0
captureId: isa-server:8080/isa;1
```

Get Message Details Status Response	
Content-Type	text/xml;charset=UTF-8
Returned element	statusMessageDetails

Example response body:

```
<?xml version="1.0" encoding="UTF-8">
<isa:statusMessageDetails>
    <isa:captureId>localhost:8080/isa;3</ns3:captureId>
    <isa:recordId>
        <pluginType>mpc</pluginType>
        <pluginId>2122219025</pluginId>
    </ns3:recordId>
    <messageId>
        <pluginId>FLOW:4097.17.3.4206064828416.0.0.0</pluginId>
        <pluginType>g10</pluginType>
    </messageId>
    <uploadPercentage>0</uploadPercentage>
</isa:statusMessageDetails>
```

2.9 MPC

Multiprotocol correlation functionality groups distinct call records with similar parameters into a single records to be analyzed as a whole.

2.9.1 Merge

Merge MPC records.

Merge Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/mpc/merge	{baseURL}/InstanceId/mpc/merge
Method	POST	
Request headers (required)	captureId	
Request parameter (required)	recordId	

Example request:

```
POST {baseURL}/mpc/merge
captureId: isa-server:8080/isa;1
recordId: mpc:760301274, mpc:1840454456
```

Merge Response	
Content-Type	text/xml;charset=UTF-8
Returned element	none

Example response body:

A successful body response has no payload.

2.9.2 Reset

Reset MPC records.

Reset Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/mpc/reset	{baseURL}/InstanceId/mpc/reset
Mapping	{baseURL}/mpc/reset	
Method	POST	
Request headers (required)	captureId	
Request parameter (required)	recordId	

Example request:

```
POST {baseURL}/mpc/reset

captureId: isa-server:8080/isa;1
recordId: mpc:760301274, mpc:1840454456
```

Reset Response	
Content-Type	text/xml;charset=UTF-8
Returned element	none

Example response body:

A successful body response has no payload.

2.9.3 Split

Split MPC records.

Split Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/mpc/split	{baseURL}/InstanceId/mpc/split
Method	POST	
Request headers (required)	captureId	
Request parameter (required)	recordId	

Example request:

```
POST {baseURL}/mpc/split
captureId: isa-server:8080/isa;1
recordId: mpc:760301274, mpc:1840454456
```

Split Response	
Content-Type	text/xml;charset=UTF-8
Returned element	none

Example response body:

A successful body response has no payload.

2.9.4 Retrieve

Retrieve MPC records.

Retrieve Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/mpc/retrieve	{baseURL}/InstanceId/mpc/retrieve
Method	POST	
Request headers (required)	captureId	
Request parameter (required)	recordId	

Example request:

```
POST {baseURL}/mpc/retrieve
captureId: isa-server:8080/isa;1
recordId: mpc:760301274, mpc:1840454456
```

Retrieve Response	
Content-Type	text/xml;charset=UTF-8
Returned element	none

Example response body:

A successful body response has no payload.

2.9.5 Process Limited

Process limited MPC.

Process Limited Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/mpc/process_limited	{baseURL}/InstanceId/mpc/process_limited
Method	POST	
Request headers (required)	captureId	
Request parameter (required)	recordId	

Example request:

```
POST {baseURL}/mpc/process_limited

captureId: isa-server:8080/isa;1
recordId: mpc:760301274, mpc:1840454456
```

Process Limited Response	
Content-Type	text/xml;charset=UTF-8
Returned element	none

Example response body:

A successful body response has no payload.

2.9.6 Process Full

Process full MPC.

Process Full Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/mpc/process_full	{baseURL}/InstanceId/mpc/process_full
Method	POST	
Request headers (required)	captureId	
Request parameter (required)	recordId	

Example request:

```
POST {baseURL}/mpc/process_full
captureId: isa-server:8080/isa;1
recordId: mpc:760301274, mpc:1840454456
```

Process Full Response	
Content-Type	text/xml;charset=UTF-8
Returned element	none

Example response body:

A successful body response has no payload.

2.9.7 Stop

Stop MPC.

Stop Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/mpc/stop	{baseURL}/InstanceId/mpc/stop
Method	POST	
Request headers (required)	captureId	
Request parameter (required)	none	

Example request:

```
POST {baseURL}/mpc/stop

captureId: isa-server:8080/isa;1
```

Stop Response	
Content-Type	text/xml;charset=UTF-8
Returned element	none

Example response body:

A successful body response has no payload.

2.10 PREFERENCE

Preference mapping provide access to set and get user preferences and use the URL pattern {baseURL}/preference/{preferenceName}.

2.10.1 Get User Preferences

Get user preferences.

Get User Preferences Request	
Mapping	{baseURL}/preference/{preferenceName}
Method	GET
Request headers (required)	username
Path variables (required)	preferenceName
Request Parameters (optional)	useDefault

Example request:

```
GET {baseURL}/preference/ladderNodeOrdering
username: user
```

Get User Preferences Response	
Content-Type	text/xml;charset=UTF-8
Returned element	preference values

Example response body:

```
["SUBSCRIBER IP Node","Generic - OnDemand","eNodeB","NodeB","BSC","HNB-
GW","BSS","SSP","STP/SSP","STP","SCP","ISDN","RNC","MME","SGSN","GGSN","GSN
Network","MMS","HSS","EIR","MSC","3G MSC","MGC","SBC","SGW","MGW","SIP-R","SIP-P","SIP-EP","P-
CSCF","I-CSCF","S-CSCF","AF","AS","ePCF","HSGW","PDN-
GW","PCEF","BBERF","PCRF","AAA","DNS","CDF","OCS","IP Node","IP Cloud"]
```

2.10.2 Set User Preferences

Set user preferences.

Set User Preferences Request	
Mapping	{baseURL}/preference/{preferenceName}
Method	POST
Request headers (required)	username
Path variables (required)	preferenceName
Request Parameters (required)	preference

Example request:

```
POST {baseURL}/preference/ladderNodeOrdering
username: user
```

Set User Preferences Response	
Content-Type	text/xml;charset=UTF-8
Returned element	status

Example response body:

```
{"status":"ok"}
```

2.11 STATUS

Status mappings provide feedback on current state of various ISA processes.

2.11.1 Get Capture Status

Get capture status.

Get Capture Status Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/status/getCaptureStatus	{baseURL}/InstanceId/status/getCaptureStatus
Method	GET	
Request headers (required)	captureId	

Example request:

```
GET {baseURL}/status/getCaptureStatus
captureId: isa-server:8080/isa;1
```

Get Capture Status Response	
Content-Type/Encoding	text/xml;charset=UTF-8/gzip
Returned element	captureStatusInfos

Example response body:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<captureStatusInfos>
    <captureStatus>
        <matchingRecords>0</matchingRecords>
        <unpausable>true</unpausable>
        <realtimeReached>true</realtimeReached>
        <pausedOnTime>true</pausedOnTime>
        <maxTime>01/01/1970 00:00:00.000000000 +0000</maxTime>
        <state>COMPLETE</state>
    </captureStatus>
</captureStatusInfos>
```

2.11.2 Get Message Status

Get message status.

Get Message Status Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/status/getMessageStatus/{recordId}	{baseURL}/InstanceId/status/getMessageStatus/{recordId}
Method	GET	
Request headers (required)	captureId	
Path variable (required)	recordId	

Example request:

```
GET {baseURL}/status/getMessageStatus/{recordId}
captureId: isa-server:8080/isa;1
```

Get Message Status Response	
Content-Type/Encoding	text/xml;charset=UTF-8/gzip
Returned element	messageRetrievalStatusInfos

Example response body:

```
<?xml version="1.0" encoding="UTF-8">
<msgRetrievalStatusInfos>
    <msgRetrievalStatus xsi:nil="true"/>
</msgRetrievalStatusInfos>
```

2.11.3 Get Status Task

Get Advanced Search status.

Get Status Task Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/status/getStatusTask	{baseURL}/InstanceId/status/getStatusTask
Method	GET	
Request headers (required)	captureId	
Path variable (required)	statusMode categoryMode	

Example request:

```
GET {baseURL}/status/getStatusTask
captureId: isa-server:8080/isa;1
```

`statusMode: CLIENT`

`categoryMode: ADVANCED`

Get Status Task Response	
Content-Type/Encoding	text/xml;charset=UTF-8/gzip
Returned element	RootTask

Example response body:

```
<?xml version="1.0" ?>
<RootTask description="" id="994578841" isTopLevelTask="false" name="DefaultTask" progress="0.0"
state="RUNNING" taskAction="NONE" taskType="ROOT">
    <attributes>
        <Attribute name="CaptureId" value="carlova-vm1:11300/isa;36"/>
    </attributes>
    <kpis/>
    <states/>
    <InitialCaptureTask description="0 out of 0 segment(s)" id="INITIAL_CAPTURE_-1987694962"
isTopLevelTask="true" name="Session Capture" progress="100.0" state="COMPLETED"
taskAction="NONE" taskType="INITIAL_CAPTURE">
```

```
        <attributes>
            <Attribute name="CaptureId" value="carlova-vm1:11300/isa;36"/>
        </attributes>
        <kpis>
            <Attribute name="Searched(Segment)" value="0"/>
            <Attribute name="Not Searched(Segment)" value="0"/>
            <Attribute name="Total(Segment)" value="0"/>
            <Attribute name="Searched(Slice)" value="0"/>
            <Attribute name="Not Searched(Slice)" value="0"/>
            <Attribute name="Total(Slice)" value="0"/>
            <Attribute name="Searched(TrackingEngine)" value="0"/>
            <Attribute name="Not Searched(TrackingEngine)" value="0"/>
            <Attribute name="Total(TrackingEngine)" value="0"/>
            <Attribute name="RecordsMatched" value="0"/>
            <Attribute name="SegmentMatchesDisk" value="0"/>
            <Attribute name="SegmentMatchesOpenRecords" value="0"/>
            <Attribute name="SrQuery Requested" value="1"/>
            <Attribute name="SrQuery Completed" value="1"/>
        </kpis>
        <states>
            <Attribute name="Running" value="0"/>
            <Attribute name="Error" value="0"/>
            <Attribute name="Completed" value="1"/>
        </states>
        <InitialCaptureServiceTask description="0 out of 0 segment(s)"
id="INIIAL_CAPTURE_SERVICE_-301384349" isTopLevelTask="false" name="g206" progress="100.0"
state="COMPLETED" taskAction="STOP" taskType="INIIAL_CAPTURE_SERVICE">
            <attributes>
                <Attribute name="CaptureId" value="carlova-vm1:11300/isa;36"/>
                <Attribute name="Id" value="4129"/>
                <Attribute name="URL" value="/capture?probeId=4129"/>
                <Attribute name="ProbeType" value="g10"/>
                <Attribute name="ProbeName" value="g206"/>
                <Attribute name="ProtocolVersion" value="6"/>
            </attributes>
            <kpis>
                <Attribute name="Searched(Segment)" value="0"/>
                <Attribute name="Not Searched(Segment)" value="0"/>
                <Attribute name="Total(Segment)" value="0"/>
                <Attribute name="Searched(Slice)" value="0"/>
                <Attribute name="Not Searched(Slice)" value="0"/>
                <Attribute name="Total(Slice)" value="0"/>
                <Attribute name="Searched(TrackingEngine)" value="0"/>
                <Attribute name="Not Searched(TrackingEngine)" value="0"/>
                <Attribute name="Total(TrackingEngine)" value="0"/>
                <Attribute name="RecordsMatched" value="0"/>
                <Attribute name="SegmentMatchesDisk" value="0"/>
                <Attribute name="SegmentMatchesOpenRecords" value="0"/>
                <Attribute name="SrQuery Requested" value="1"/>
                <Attribute name="SrQuery Completed" value="1"/>
            </kpis>
            <states/>
        </InitialCaptureServiceTask>
    </InitialCaptureTask>
    <MpcCaptureTask description="0 out of 0 record request(s)" id="MPC_CAPTURE_-1987694962"
isTopLevelTask="true" name="MPC" progress="0.0" state="IDLE" taskAction="NONE"
taskType="MPC_CAPTURE">
        <states>
            <Attribute name="Idle" value=""/>
        </states>
        <attributes>
            <Attribute name="CaptureId" value="carlova-vm1:11300/isa;36"/>
            <Attribute name="isMpcSrQueryTaskKept" value="false"/>
        </attributes>
        <kpis/>
    </MpcCaptureTask>
    <MessageCaptureTask description="0 out of 0 detail request(s)" id="MESSAGE_CAPTURE_-
1987694962" isTopLevelTask="true" name="Message Capture" probeName="" progress="0.0"
state="IDLE" taskAction="NONE" taskType="MESSAGE_CAPTURE">
```

```
        <states>
            <Attribute name="Idle" value="1"/>
        </states>
        <attributes>
            <Attribute name="CaptureId" value="carlova-vm1:11300/isa;36"/>
        </attributes>
        <kpis/>
    </MessageCaptureTask>
    <FlowExpansionTask description="0 out of 0 detail request(s)" id="FLOW_EXPANSION_-
1987694962" isTopLevelTask="true" name="Flow Expansion" probeName="" progress="0.0" state="IDLE"
taskAction="NONE" taskType="FLOW_EXPANSION">
        <states>
            <Attribute name="Idle" value="1"/>
        </states>
        <attributes>
            <Attribute name="CaptureId" value="carlova-vm1:11300/isa;36"/>
        </attributes>
        <kpis/>
    </FlowExpansionTask>
    <ExportTask description="0 out of 0 export task(s)" id="EXPORT_-1987694962"
isTopLevelTask="true" name="Export" progress="0.0" state="IDLE" taskAction="NONE"
taskType="EXPORT">
        <attributes>
            <Attribute name="CaptureId" value="carlova-vm1:11300/isa;36"/>
        </attributes>
        <kpis/>
        <states/>
    </ExportTask>
    <MediaAnalysisTask description="0 out of 0 detail request(s)" id="MEDIA_ANALYSIS_-
1987694962" isTopLevelTask="true" name="Media Analysis" probeName="" progress="0.0" state="IDLE"
taskAction="NONE" taskType="MEDIA_ANALYSIS">
        <states>
            <Attribute name="Idle" value="1"/>
        </states>
        <attributes>
            <Attribute name="CaptureId" value="carlova-vm1:11300/isa;36"/>
        </attributes>
        <kpis/>
    </MediaAnalysisTask>
    <EntireTimeCaptureTask description="0 out of 0 record request(s)" id="ENTIRE_TIME_CAPTURE_-
1987694962" isTopLevelTask="true" name="Retrieve Entire Time Range" progress="0.0" state="IDLE"
taskAction="NONE" taskType="ENTIRE_TIME_CAPTURE">
        <states>
            <Attribute name="Idle" value=""/>
        </states>
        <attributes>
            <Attribute name="CaptureId" value="carlova-vm1:11300/isa;36"/>
        </attributes>
        <kpis/>
    </EntireTimeCaptureTask>
    <BufferTask description="0 out of 1000 record(s)" id="BUFFER_-1987694962"
isTopLevelTask="true" name="Buffer" progress="0.0" state="AVAILABLE" taskAction="NONE"
taskType="BUFFER">
        <attributes>
            <Attribute name="CaptureId" value="carlova-vm1:11300/isa;36"/>
        </attributes>
        <kpis>
            <Attribute name="Total Limit" value="1000"/>
            <Attribute name="Current Total Buffer" value="0"/>
        </kpis>
        <states>
            <Attribute name="Available" value="2"/>
            <Attribute name="Unavailable" value="0"/>
        </states>
        <OverallRecordBufferTask description="0 out of 1000 record(s)"
id="OVERALL_RECORD_BUFFER_-1987694962" isTopLevelTask="false" name="Overall Record Buffer"
progress="0.0" state="AVAILABLE" taskAction="NONE" taskType="OVERALL_RECORD_BUFFER">
            <attributes>
                <Attribute name="CaptureId" value="carlova-vm1:11300/isa;36"/>
```

```
            </attributes>
            <kpis>
                <Attribute name="Current Total Record Buffer" value="0"/>
                <Attribute name="Total Record Buffer Limit" value="1000"/>
            </kpis>
            <states/>
        </OverallRecordBufferTask>
        <UserPlanePDUBufferTask description="0 out of 5000 pdu(s)" id="USERPLANE_PDU_BUFFER_-
1987694962" isTopLevelTask="false" name="Flow PDUs Buffer" progress="0.0" state="AVAILABLE"
taskAction="NONE" taskType="USERPLANE_PDU_BUFFER">
            <attributes>
                <Attribute name="CaptureId" value="carlova-vm1:11300/isa;36"/>
            </attributes>
            <kpis>
                <Attribute name="Current Flow PDU Buffer(per session)" value="0"/>
                <Attribute name="Flow PDU Buffer Limit(per session)" value="5000"/>
            </kpis>
            <states/>
        </UserPlanePDUBufferTask>
    </BufferTask>
</RootTask>
```

2.12 TOPOLOGY

Topology mapping provide data and management of monitored objects and nodes.

2.12.1 Get Probes

Get connected probes.

Get Probes Request	
Mapping	{baseURL}/topology/probes
Method	GET
Request Parameters (optional)	groups options pluginType

Example request:

```
GET {baseURL}/topology/probes

groups=nodeGroup:1
pluginType=G10
```

Get Probes Response	
Content-Type/Encoding	text/xml;charset=UTF-8/gzip
Returned element	monitoredObjects

Example response body:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<isa:monitoredObjects>
   <monitoredObject>
        <displayName>g107</displayName>
        <groupNames/>
        <monitoredIds>
            <monitoredId>
                <pluginId>Probe:4097</pluginId>
                <pluginType>g10</pluginType>
            </monitoredId>
        </monitoredIds>
        <networkType>Probe</networkType>
        <providers>
            <provider>
                <apiVersion>5</apiVersion>
                <probeDisplayName>g107</probeDisplayName>
                <probeId>4097</probeId>
                <type>g10</type>
                <uri>g10://134.64.37.78:11109</uri>
            </provider>
        </providers>
    </monitoredObject>
</isa:monitoredObjects>
```

2.12.2 Get Monitored Nodes

Get monitored nodes.

Get Monitored Nodes Request	
Mapping	{baseURL}/topology/nodes
Method	GET
Request Parameters (optional)	groups pluginType

Example request:

```
GET {baseURL}/topology/nodes

pluginType=g10
```

Get Monitored Nodes Response	
Content-Type/Encoding	text/xml;charset=UTF-8/gzip
Returned element	monitoredNodes

Example response body:

```
<?xml version="1.0" encoding="UTF-8">
<isa:monitoredNodes>
    <monitoredNode>
        <id>
            <pluginId>NODE:15831</pluginId>
            <pluginType>g10</pluginType>
        </id>
        <name>SGSN/158.230.242.65</name>
        <type>SGSN</type>
        <ipRange>158.230.242.65,158.230.242.97</ipRange>
        <pluginIdentifiers/>
        <associatedNodes/>
    </monitoredNode>
    <monitoredNode>
        <id>
            <pluginId>NODE:15847</pluginId>
            <pluginType>g10</pluginType>
        </id>
        <name>SGSN/213.191.235.10</name>
        <type>SGSN</type>
        <ipRange>213.191.235.10,213.191.235.38</ipRange>
        <pluginIdentifiers/>
        <associatedNodes/>
    </monitoredNode>
    [...]
</isa:monitoredNodes>
```

2.12.3 Get Monitored Links

Get monitored links.

Get Monitored Links Request	
Mapping	{baseURL}/topology/links
Method	POST
Request Parameters (optional)	pluginType pluginIds

Example request:

```
GET {baseURL}/topology/links

pluginType: g10
```

Get Monitored Links Response	
Content-Type/Encoding	text/xml;charset=UTF-8/gzip
Returned element	monitoredNodes

Example response body:

```
<?xml version="1.0" encoding="UTF-8">
<isa:monitoredNodes>
    <monitoredNode>
        <id>
            <pluginId>NODE:15831</pluginId>
            <pluginType>g10</pluginType>
        </id>
        <name>SGSN/158.230.242.65</name>
        <type>SGSN</type>
        <ipRange>158.230.242.65,158.230.242.97</ipRange>
        <pluginIdentifiers/>
        <associatedNodes/>
    </monitoredNode>
    <monitoredNode>
        <id>
            <pluginId>NODE:15847</pluginId>
            <pluginType>g10</pluginType>
        </id>
        <name>SGSN/213.191.235.10</name>
        <type>SGSN</type>
        <ipRange>213.191.235.10,213.191.235.38</ipRange>
        <pluginIdentifiers/>
        <associatedNodes/>
    </monitoredNode>
    [...]
</isa:monitoredNodes>
```

2.12.4 Get Monitored G10 Nodes

Get monitored G10 nodes.

Get Monitored G10 Nodes Request	
Mapping	{baseURL}/topology/g10/nodes
Method	GET
Request Parameters (optional)	groups

Example request:

```
GET {baseURL}/topology/g10/nodes

groups=nodegroup:1
```

Get Monitored G10 Nodes Response	
Content-Type/Encoding	text/xml;charset=UTF-8/gzip
Returned element	monitoredNodes

Example response body:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<isa:monitoredNodes>
    <monitoredNode>
        <id>
            <pluginId>NODE:15887</pluginId>
            <pluginType>g10</pluginType>
        </id>
        <name>GGSN/216.155.172.225</name>
        <type>GGSN</type>
        <ipRange>216.155.172.225,216.155.172.226,216.155.172.236</ipRange>
        <pluginIdentifiers/>
        <associatedNodes/>
    </monitoredNode>
    <monitoredNode>
        <id>
            <pluginId>NODE:16071</pluginId>
            <pluginType>g10</pluginType>
        </id>
        <name>GGSN/138.188.2.1</name>
        <type>GGSN</type>
        <ipRange>138.188.2.1</ipRange>
        <pluginIdentifiers/>
        <associatedNodes/>
    </monitoredNode>
    [...]
</isa:monitoredNodes>
```

2.12.5 Get Monitored SPI Nodes

Get monitored SPI nodes.

Get Monitored SPI Nodes Request	
Mapping	{baseURL}/topology/spi/nodes
Method	GET
Request Parameters (optional)	groups

Example request:

```
GET {baseURL}/topology/spi/nodes

groups= nodegroup:1
```

Get Monitored SPI Nodes Response	
Content-Type/Encoding	text/xml;charset=UTF-8/gzip
Returned element	monitoredNodes

Example response body:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<isa:monitoredNodes>
    <monitoredNode>
        <id>
            <pluginId>NODE:15887</pluginId>
            <pluginType>g10</pluginType>
        </id>
        <name>GGSN/216.155.172.225</name>
        <type>GGSN</type>
        <ipRange>216.155.172.225,216.155.172.226,216.155.172.236</ipRange>
        <pluginIdentifiers/>
        <associatedNodes/>
    </monitoredNode>
    <monitoredNode>
        <id>
            <pluginId>NODE:16071</pluginId>
            <pluginType>spi</pluginType>
        </id>
        <name>GGSN/138.188.2.1</name>
        <type>GGSN</type>
        <ipRange>138.188.2.1</ipRange>
        <pluginIdentifiers/>
        <associatedNodes/>
    </monitoredNode>
    [...]
</isa:monitoredNodes>
```

2.12.6 Get Monitored Nodes By Type

Get monitored nodes by type.

Get Monitored SPI Nodes Request	
Mapping	{baseURL}/topology//nodes
Method	GET
Request Parameters (optional)	groups

Example request:

```
GET {baseURL}/topology/spi/nodes

groups= nodegroup:1
```

Get Monitored SPI Nodes Response	
Content-Type/Encoding	text/xml;charset=UTF-8/gzip
Returned element	monitoredNodes

Example response body:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<isa:monitoredNodes>
    <monitoredNode>
        <id>
            <pluginId>NODE:15887</pluginId>
            <pluginType>g10</pluginType>
        </id>
        <name>GGSN/216.155.172.225</name>
        <type>GGSN</type>
        <ipRange>216.155.172.225,216.155.172.226,216.155.172.236</ipRange>
        <pluginIdentifiers/>
        <associatedNodes/>
    </monitoredNode>
    <monitoredNode>
        <id>
            <pluginId>NODE:16071</pluginId>
            <pluginType>spi</pluginType>
        </id>
        <name>GGSN/138.188.2.1</name>
        <type>GGSN</type>
        <ipRange>138.188.2.1</ipRange>
        <pluginIdentifiers/>
        <associatedNodes/>
    </monitoredNode>
    [...]
</isa:monitoredNodes>
```

2.12.7 Get Monitored Node Groups

Get monitored node groups.

Get Monitored Node Groups Request	
Mapping	{baseURL}/topology/nodeGroups
Method	GET
Request Parameters (optional)	pluginType

Example request:

```
GET {baseURL}/topology/nodeGroups

pluginType = nodeGroup:1
```

Get Monitored Node Groups Response	
Content-Type/Encoding	text/xml;charset=UTF-8/gzip
Returned element	monitoredObjectGroups

Example response body:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<isa:monitoredObjectGroups>
    <monitoredObjectGroup>
        <groupIdentifier>
            <pluginId>NODEGROUP:5</pluginId>
            <pluginType>g10</pluginType>
        </groupIdentifier>
        <groupName>blah</groupName>
        <otherPluginIds/>
    </monitoredObjectGroup>
    <monitoredObjectGroup>
        <groupIdentifier>
            <pluginId>NODEGROUP:0</pluginId>
            <pluginType>g10</pluginType>
        </groupIdentifier>
        <groupName>ungrouped</groupName>
        <otherPluginIds/>
    </monitoredObjectGroup>
</isa:monitoredObjectGroups>
```

2.12.8 Get Monitored SPI Node Groups

Get monitored SPI node groups.

Get Monitored SPI Node Groups Request	
Mapping	{baseURL}/topology/spi/nodeGroups
Method	GET

Example request:

```
GET {baseURL}/topology/spi/nodeGroups
```

Get Monitored SPI Node Groups Response	
Content-Type/Encoding	text/xml;charset=UTF-8/gzip
Returned element	monitoredObjectGroups

Example response body:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<isa:monitoredObjectGroups>
    <monitoredObjectGroup>
        <groupIdentifier>
            <pluginId>NODEGROUP:5</pluginId>
            <pluginType>spi</pluginType>
        </groupIdentifier>
        <groupName>blah</groupName>
        <otherPluginIds/>
    </monitoredObjectGroup>
    <monitoredObjectGroup>
        <groupIdentifier>
            <pluginId>NODEGROUP:0</pluginId>
            <pluginType>spi</pluginType>
        </groupIdentifier>
        <groupName>ungrouped</groupName>
        <otherPluginIds/>
    </monitoredObjectGroup>
</isa:monitoredObjectGroups>
```

2.12.9 Get Monitored G10 Node Groups

Get monitored G10 node groups.

Get Monitored G10 Node Groups Request	
Mapping	{baseURL}/topology/g10/nodeGroups
Method	GET

Example request:

```
GET {baseURL}/topology/g10/nodeGroups
```

Get Monitored G10 Node Groups Response	
Content-Type/Encoding	text/xml;charset=UTF-8/gzip
Returned element	monitoredObjectGroups

Example response body:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<isa:monitoredObjectGroups>
    <monitoredObjectGroup>
        <groupIdentifier>
            <pluginId>NODEGROUP:5</pluginId>
            <pluginType>g10</pluginType>
        </groupIdentifier>
        <groupName>test-group</groupName>
        <otherPluginIds/>
    </monitoredObjectGroup>
    <monitoredObjectGroup>
        <groupIdentifier>
            <pluginId>NODEGROUP:0</pluginId>
            <pluginType>g10</pluginType>
        </groupIdentifier>
        <groupName>ungrouped</groupName>
        <otherPluginIds/>
    </monitoredObjectGroup>
</isa:monitoredObjectGroups>
```

2.12.10 Get Point Codes

Get point codes.

Get Point Codes Request	
Mapping	{baseURL}/topology/pointCodes
Method	GET
Request Parameters (optional)	names

Example request:

```
GET {baseURL}/topology/pointCodes
```

Get Point Codes Response	
Content-Type/Encoding	text/xml;charset=UTF-8/gzip
Returned element	pointcodes

Example response body:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<isa:pointcodes>
    <pointcode>
        <id>SIGTRAN_NODE/001-100-001</id>
        <name>SIGTRAN_NODE/001-100-001</name>
        <networkIndicator>Natl</networkIndicator>
        <pointcode>001-100-001</pointcode>
        <pointcodeBytes>CAFkAQ==</pointcodeBytes>
        <protocol>ANSI</protocol>
        <type>SIGTRAN_NODE</type>
    </pointcode>
    <pointcode>
        <id>SIGTRAN_NODE/001-110-001</id>
        <name>SIGTRAN_NODE/001-110-001</name>
        <networkIndicator>Natl</networkIndicator>
        <pointcode>001-110-001</pointcode>
        <pointcodeBytes>CAFuAQ==</pointcodeBytes>
        <protocol>ANSI</protocol>
        <type>SIGTRAN_NODE</type>
    </pointcode>
    <pointcode>
        <id>SIGTRAN_NODE/0-000-1</id>
        <name>SIGTRAN_NODE/0-000-1</name>
        <networkIndicator>Natl</networkIndicator>
        <pointcode>0-000-1</pointcode>
        <pointcodeBytes>KAAAAQ==</pointcodeBytes>
        <protocol>ITU</protocol>
        <type>SIGTRAN_NODE</type>
    </pointcode>
</isa:pointcodes>
```

2.12.11 Get Probe Groups

Get probe groups.

Get Probe Groups Request	
Mapping	{baseURL}/topology/probeGroups
Method	GET

Example request:

```
GET {baseURL}/topology/probeGroups
```

Get Probe Groups Response	
Content-Type/Encoding	text/xml;charset=UTF-8/gzip
Returned element	monitoredObjectGroups

Example response body:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<isa:monitoredObjectGroups>
    <monitoredObjectGroup>
        <groupIdentifier>
            <pluginId>ProbeGroup:0</pluginId>
            <pluginType>g10</pluginType>
        </groupIdentifier>
        <groupName>ungrouped</groupName>
        <otherPluginIds/>
    </monitoredObjectGroup>
</isa:monitoredObjectGroups>
```

2.12.12 Get Monitored Objects Groups

Get monitored objects groups.

Get Monitored Objects Groups Request	
Mapping	{baseURL}/topology/groups
Method	GET

Example request:

```
GET {baseURL}/topology/groups
```

Get Monitored Objects Groups Response	
Content-Type/Encoding	text/xml;charset=UTF-8/gzip
Returned element	monitoredObjectGroups

Example response body:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<isa:monitoredObjectGroups>
    <monitoredObjectGroup>
        <groupIdentifier>
            <pluginId>ProbeGroup:0</pluginId>
            <pluginType>g10</pluginType>
        </groupIdentifier>
        <groupName>ungrouped</groupName>
        <otherPluginIds/>
    </monitoredObjectGroup>
</isa:monitoredObjectGroups>
```

2.12.13 Get Node Types

Get node types.

Get Node Types Request	
Mapping	{baseURL}/topology/nodeTypes
Method	GET

Example request:

```
GET {baseURL}/topology/nodeTypes
```

Get Node Types Response	
Content-Type	text/xml;charset=UTF-8
Returned element	node types

Example response body:

```
["TPF","HSS","OCS","SGSN","SIGTRAN_NODE","GGSN","OCF","AAA","SCP","BSS","BBERF","DNS","ISDN","SG
W","MSRP","AS","MGW","BSC","I-CSCF","SIP-P","ePCF","SIP-R","IT Server","MSC","P-
CSCF","MRF","CTF","IP Node","HNB-GW","RNC","AF","MGC","STP","NodeB","SBC","HSGW","MME","PDN-
GW","Transparent Network Device","SIP-EP","eNodeB","CDF","MMS","PCRF","IP Cloud","PDF","EIR","3G
MSC","MRFP","GSN Network","Generic - OnDemand","MRFC","S-CSCF","CRF","STP/SSP","SSP","PCEF"]
```

2.12.14 Get Links

Get links.

Get Links Request	
Mapping	{baseURL}/topology/g10/links
Method	GET

Example request:

```
GET {baseURL}/topology/g10/links
```

Get Links Response	
Content-Type	text/xml;charset=UTF-8
Returned element	linkList

Example response body:

```
<?xml version="1.0" encoding="UTF-8">
<isa:linkList>
    <links>
        <id>49</id>
        <name>test</name>
        <serverNodeId>15694</serverNodeId>
        <clientNodeId>15716</clientNodeId>
        <protocol>Sctp</protocol>
    </links>
    <links>
        <id>51</id>
        <name>test2</name>
        <serverNodeId>15694</serverNodeId>
        <clientNodeId>15700</clientNodeId>
        <protocol>Sctp</protocol>
    </links>
    <links>
        <id>52</id>
        <name>test3</name>
        <serverNodeId>15702</serverNodeId>
        <clientNodeId>15716</clientNodeId>
        <protocol>Sctp</protocol>
    </links>
    <links>
        <id>36</id>
        <name>test logical link</name>
        <serverNodeId>15660</serverNodeId>
        <clientNodeId>15658</clientNodeId>
        <protocol>Tcp</protocol>
    </links>
    <links>
        <id>53</id>
        <name>MMESGW</name>
        <serverNodeId>15694</serverNodeId>
        <clientNodeId>15696</clientNodeId>
        <protocol>Sctp</protocol>
    </links>
    <links>
        <id>37</id>
        <name>MME-eNodeB</name>
        <serverNodeId>15702</serverNodeId>
        <clientNodeId>15706</clientNodeId>
        <protocol>Sctp</protocol>
    </links>
    <links>
        <id>50</id>
        <name>test1</name>
        <serverNodeId>15694</serverNodeId>
        <clientNodeId>15702</clientNodeId>
        <protocol>Sctp</protocol>
    </links>
</isa:linkList>
```

2.13 REMOTE SERVER REPOSITORY

Provides access to remote server repository.

2.13.1 Get Record Count by User

Get the number of records for a user.

Get Record Count by User Request	
Mapping	{baseURL}/repository/getRecordCountByUser
Method	GET
Request Parameters (required)	count filterBy filterValue hasAdminRole index loginUserId selectedUserId sorterBy sorterOrder

Example request:

```
GET {baseURL}/repository/getRecordCountByUser

selectedUserId: user1
loginUserId: admin
hasAdminRole: true
```

Get Record Count by User Response	
Content-Type	text/text;charset=ISO-8859-1
Returned element	number of records

Example response body:

```
152
```

2.13.2 Get Record Count by Profile

Get the number of records for a VIP profile.

Get Record Count by Profile Request	
Mapping	{baseURL}/repository/getRecordCountByProfile
Method	GET
Request Parameters (required)	count filterBy filterValue hasVipRole index loginUserId profileName sorterBy sorterOrder

Example request:

```
GET {baseURL}/repository/getRecordCountByProfile

profileName: vip1
loginUserId: admin
hasVipRole: true
```

Get Record Count by Profile Response	
Content-Type	text/text;charset=ISO-8859-1
Returned element	number of records

Example response body:

```
85
```

2.13.3 Get Capture List by Ids

Get the capture list for a particular captureId(s).

Get Capture List by Ids Request		
	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/repository/getCaptureListbyIds	{baseURL}/InstanceId/repository/getCaptureListbyIds
Method	GET	
Request Parameters (required)	captureId	

Example request:

```
GET {baseURL}/repository/getCaptureListbyIds

captureId: isa-server:8080/isa;1
```

Get Capture List by Ids Response	
Content-Type	text/xml;charset=ISO-8859-1
Returned element	captureSessionRecords

Example response body:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><captureSessionRecords>
<record recordID="1">
    <entity name="ID" type="LONG">
        <instance>
            <attribute>
                <name>value</name>
                <value>1</value>
            </attribute>
        </instance>
    </entity>
    <entity name="Name" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value>admin</value>
            </attribute>
```

```
        </instance>
    </entity>
    <entity name="File Name" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value>test full MPC.isa</value>
            </attribute>
        </instance>
    </entity>
    <entity name="File Size (KB)" type="FLOAT">
        <instance>
            <attribute>
                <name>value</name>
                <value>45.209</value>
            </attribute>
        </instance>
    </entity>
    <entity name="Created" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value>1350982622000</value>
            </attribute>
        </instance>
    </entity>
    <entity name="Sharing" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value>PRIVATE</value>
            </attribute>
        </instance>
    </entity>
    <entity name="Comment" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value></value>
            </attribute>
        </instance>
    </entity>
</record>
<record recordID="3">
    <entity name="ID" type="LONG">
        <instance>
            <attribute>
                <name>value</name>
                <value>3</value>
            </attribute>
        </instance>
    </entity>
    <entity name="Name" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value>admin</value>
            </attribute>
        </instance>
    </entity>
    <entity name="File Name" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value>expand.isa</value>
            </attribute>
        </instance>
    </entity>
    <entity name="File Size (KB)" type="FLOAT">
```

```
        <instance>
            <attribute>
                <name>value</name>
                <value>22.795</value>
            </attribute>
        </instance>
    </entity>
    <entity name="Created" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value>1351010158000</value>
            </attribute>
        </instance>
    </entity>
    <entity name="Sharing" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value>PRIVATE</value>
            </attribute>
        </instance>
    </entity>
    <entity name="Comment" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value></value>
            </attribute>
        </instance>
    </entity>
</record>
</captureSessionRecords>
```

2.13.4 Get Capture List by User

Get the capture list for a particular user.

Get Capture List by User Request	
Mapping	{baseURL}/repository/getCaptureListByUser
Method	GET
Request Parameters (required)	count hasAdminRole index loginUserId selectedUserId
Request Parameters (optional)	sorterBy sorterOrder filterBy filterValue

Example request:

```
GET {baseURL}/repository/getCaptureListByUser

count:100
hasAdminRole: true
index: 0
loginUserId: admin
selectedUserId: admin
```

Get Capture List by User Response	
Content-Type	text/xml;charset=ISO-8859-1
Returned element	captureSessionRecords

Example response body:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><captureSessionRecords>
<record recordID="1">
    <entity name="ID" type="LONG">
        <instance>
            <attribute>
                <name>value</name>
                <value>1</value>
            </attribute>
        </instance>
    </entity>
    <entity name="Name" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value>admin</value>
            </attribute>
        </instance>
    </entity>
    <entity name="File Name" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value>test full MPC.isa</value>
            </attribute>
        </instance>
    </entity>
    <entity name="File Size (KB)" type="FLOAT">
        <instance>
            <attribute>
                <name>value</name>
                <value>45.209</value>
            </attribute>
        </instance>
    </entity>
    <entity name="Created" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value>1350982622000</value>
            </attribute>
        </instance>
    </entity>
    <entity name="Sharing" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value>PRIVATE</value>
            </attribute>
        </instance>
    </entity>
    <entity name="Comment" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value></value>
            </attribute>
        </instance>
    </entity>
</record>
```

```
<record recordID="3">
    <entity name="ID" type="LONG">
        <instance>
            <attribute>
                <name>value</name>
                <value>3</value>
            </attribute>
        </instance>
    </entity>
    <entity name="Name" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value>admin</value>
            </attribute>
        </instance>
    </entity>
    <entity name="File Name" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value>expand.isa</value>
            </attribute>
        </instance>
    </entity>
    <entity name="File Size (KB)" type="FLOAT">
        <instance>
            <attribute>
                <name>value</name>
                <value>22.795</value>
            </attribute>
        </instance>
    </entity>
    <entity name="Created" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value>1351010158000</value>
            </attribute>
        </instance>
    </entity>
    <entity name="Sharing" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value>PRIVATE</value>
            </attribute>
        </instance>
    </entity>
    <entity name="Comment" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value></value>
            </attribute>
        </instance>
    </entity>
</record>
</captureSessionRecords>
```

2.13.5 Get Capture List by Profile

Get the capture list for a particular VIP profile.

Get Capture List by Profile Request	
Mapping	{baseURL}/repository/getCaptureListByProfile
Method	GET
Request Parameters (required)	count filterBy filterValue hasVipRole index loginUserId profileName sorterBy sorterOrder

Example request:

```
GET {baseURL}/repository/getCaptureListByProfile

profileName: vip1
loginUserId: admin
hasVipRole: true
```

Get Capture List by Profile Response	
Content-Type	text/xml;charset=ISO-8859-1
Returned element	captureSessionRecords

Example response body:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<captureSessionRecords>
    <record recordID="2">
        <entity name="ID" type="LONG">
            <instance>
                <attribute>
                    <name>value</name>
                    <value>2</value>
                </attribute>
            </instance>
        </entity>
        <entity name="Name" type="ID_STRING">
            <instance>
                <attribute>
                    <name>value</name>
                    <value>GSM</value>
                </attribute>
            </instance>
        </entity>
        <entity name="File Name" type="ID_STRING">
            <instance>
                <attribute>
                    <name>value</name>
                    <value>artifact_00000001_20121023172416.isa</value>
                </attribute>
            </instance>
        </entity>
        <entity name="File Size (KB)" type="FLOAT">
            <instance>
                <attribute>
                    <name>value</name>
```

```
                <value>38.587</value>
            </attribute>
        </instance>
    </entity>
    <entity name="Created" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value>1350984352000</value>
            </attribute>
        </instance>
    </entity>
    <entity name="Expired" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value>1353576352000</value>
            </attribute>
        </instance>
    </entity>
    <entity name="Sharing" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value>PRIVATE</value>
            </attribute>
        </instance>
    </entity>
    <entity name="Comment" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value>100000000000001</value>
            </attribute>
        </instance>
    </entity>
</record>
<record recordID="4">
    <entity name="ID" type="LONG">
        <instance>
            <attribute>
                <name>value</name>
                <value>4</value>
            </attribute>
        </instance>
    </entity>
    <entity name="Name" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value>GSM</value>
            </attribute>
        </instance>
    </entity>
    <entity name="File Name" type="ID_STRING">
        <instance>
            <attribute>
                <name>value</name>
                <value>artifact_00000001_20121024101800.isa</value>
            </attribute>
        </instance>
    </entity>
    <entity name="File Size (KB)" type="FLOAT">
        <instance>
            <attribute>
                <name>value</name>
                <value>38.582</value>
            </attribute>
        </instance>
```

```
        </entity>
        <entity name="Created" type="ID_STRING">
            <instance>
                <attribute>
                    <name>value</name>
                    <value>1351045161000</value>
                </attribute>
            </instance>
        </entity>
        <entity name="Expired" type="ID_STRING">
            <instance>
                <attribute>
                    <name>value</name>
                    <value>1353637161000</value>
                </attribute>
            </instance>
        </entity>
        <entity name="Sharing" type="ID_STRING">
            <instance>
                <attribute>
                    <name>value</name>
                    <value>PRIVATE</value>
                </attribute>
            </instance>
        </entity>
        <entity name="Comment" type="ID_STRING">
            <instance>
                <attribute>
                    <name>value</name>
                    <value>100000000000001</value>
                </attribute>
            </instance>
        </entity>
    </record>
</ns23:captureSessionRecords>
```

2.13.6 Delete Capture Record

Deletes a capture record(s) for a given captureId(s).

Delete Capture Record Request

	When nginx is turned OFF	When nginx is turned ON (if different from, when turned OFF)
Mapping	{baseURL}/repository/deleteCaptureRecord	{baseURL}/InstanceId/repository/deleteCaptureRecord
Method	DELETE	
Request Parameters (required)	captureId	

Example request:

```
DELETE {baseURL}/repository/deleteCaptureRecord

captureId: id1, id2
```

Delete Capture Record Response	
Content-Type	text/text;charset=ISO-8859-1
Returned element	none

Example response body:

A successful body response has no payload.

2.13.7 Toggle Capture Record

Toggles a capture record(s) from public to private and vice-versa for a given captureId(s).

Toggle Capture Record Request		
	When nginx is turned OFF	**When nginx is turned ON (if different from, when turned OFF)**
Mapping	{baseURL}/repository/toggleCaptureRecord	{baseURL}/InstanceId/repository/toggleCaptureRecord
Method	POST	
Request Parameters (required)	captureId	

Example request:

```
POST {baseURL}/repository/toggleCaptureRecord

captureId: id1,id2
```

Toggle Capture Record Response	
Content-Type	text/text;charset=ISO-8859-1
Returned element	

Example response body:

A successful body response has no payload.

2.13.8 Rename Capture Record

Renames a capture record for a given captureId.

Rename Capture Record Request		
	When nginx is turned OFF	**When nginx is turned ON (if different from, when turned OFF)**
Mapping	{baseURL}/repository/renameCaptureRecord	{baseURL}/InstanceId/repository/renameCaptureRecord
Method	POST	
Request Parameters (required)	captureId renameSessionRecordStr	

Example request:

```
POST {baseURL}/repository/renameCaptureRecord

captureId: isa-server:8080/isa;1
renameSessionRecordStr: newname;
```

Rename Capture Record Response	
Content-Type	text/text;charset=ISO-8859-1
Returned element	The number of renamed records

Example response body:

2

2.13.9 Download Capture Session

Download a capture session for a given captureId.

Download Capture Session Request		
	When nginx is turned OFF	**When nginx is turned ON (if different from, when turned OFF)**
Mapping	{baseURL}/repository/downloadCaptureSession	{baseURL}/InstanceId/repository/downloadCaptureSession
Method	POST	
Request Parameters (required)	captureId	

Example request:

```
POST {baseURL}/repository/downloadCaptureSession

captureId: id1
```

Download Capture Session Response	
Content-Type	text/xml;charset=ISO-8859-1
Returned element	none

Example response body:

A successful body response has no payload.

2.13.10 Get User List

Get list of users.

Get User List Request	
Mapping	{baseURL}/repository/getUserList
Method	GET
Request Parameters (required)	hasAdminRole

Example request:

```
GET {baseURL}/repository/getUserList

hasAdminRole: true
```

Get User List Response	
Content-Type	text/xml;charset=ISO-8859-1
Returned element	userInfos

Example response body:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<userInfos
    <userInfo>
        <name>admin</name>
    </userInfo>
</userInfos>
```

2.13.11 Get Profile List

Get list of profiles.

Get Profile List Request	
Mapping	{baseURL}/repository/getProfileList
Method	GET
Request Parameters (required)	hasVipRole

Example request:

```
GET {baseURL}/repository/getProfileList

hasVipRole: true
```

Get Profile List Response	
Content-Type	text/xml;charset=ISO-8859-1
Returned element	profiles

Example response body:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
    <profileRecord>
        <name>GSM</name>
    </profileRecord>
</profiles>
```

3. APPENDIX

3.1 PARAMETER DETAILS

absoluteTime

[Boolean] Flag designating whether the time is displayed in absolute form or not.

values: true|false

allGroupNodeTypes

[Comma separated list of Strings] List of all nodes types in a ladder diagram used in cloud function.

example: allGroupNodeTypes = AAA, UNKNOWN

allRecords

[Boolean] Flag designating whether to retrieve all session records for an ISA export or not

values: true|false

areMessagesFiltered

[Boolean] Flag designating whether session messages have been filtered

values: true|false

captureId

[String] The unique identifier for a capture.

Follows the pattern: server:port/isa;captureIncrement

example: captureID = isa-server:8080/isa;1

categoryMode

[Enum] It is used to control how many category the user want to display on the GUI, two modes are provide, SIMPLE and ADVANCED.

values: SIMPLE|ADVANCED

chunkSize

[String] Number of objects to return per RESTful poll

example: chunkSize = 100

comment (RSR filter key)

[String] Session comment .

example: comment = problematic IMSI

content

[Enum] The source of data to deliver in an CSV export

values: records|messages

count

[String] Number of records to return per page.

example: count = 100

created

[String] Range of created dates expressed as follows:

startTime||endTime (times expressed in milliseconds since epoch)

example: created = 1351269254269||1435469254354

currentLadderDiagramRecord

[String] Designates currently selected record on ladder diagram.

example:currentLadderDiagramRecord = mpc:2135468764

doExportToRepository

[Boolean] Flag designating whether cancel export to repository or not

values: true|false

doFullMpc

[Boolean] Flag designating whether full MPC will be performed for an ISA export

values: true|false

doMessageRetrieval

[Boolean] Flag designating whether message retrieval will be performed for an ISA export

values: true|false

duration

[String] Time in hours for a session to run detached

example: duration = 2

endPointType

[Enum] Specifies the physical IP layer of an end point

values: IP Layer1|IP Layer2|IP Layer3|IP Layer4|UNKNOWN

endTime

[String] The last time of a query

Follows ISO-8061 format: yyyy-mm-ddTHH:mm:ss.SSS

example: endTime = 2012-10-21T06:52:00.000

exportTime

[String] The time stamp of an export

example: exportTime = 2012-10-21T06:52:00.000

exportType

[Enum] File format of exported session

values: HTML|CSV|PCAP|ISA

expression

[String] Filter search expression to validate.

example: expression = 'GTPv2':('Response Code' = 'GTPv2-C;;4/RAT Changed from 3GPP to Non-3GPP')

fileSize

[Integer] The maximum amount of data (in the units specified by fileSizeUnits) to export.

This parameter is only used when the parameter withUserPlanePDUs is specified.

Values: 1 - 2,147,483,647

example: fileSize = 20480

fileSizeUnits

[String] The unit to apply to fileSize.

This parameter is only used when the parameter withUserPlanePDUs is specified.

Values: MB|GB

example: fileSizeUnits = GB

file name

[String] file name

example: file name = session name

file size

[String] size range of a file in KB

example: file size = 10.8||100.8

filterBy

[String] Field to limit search by

Used in conjunction with filterValue as follows:
filterBy ={key}&filterValue ={value}

where key/value pairs are:

Created	session create time
File Name	session name
Comment	session comments
File Size	session size

See parameter details for more detail on each key.

example: filterBy=created & filterValue=2012-10-21T06:52:00.000

filterExpression

Overview

The ISA filter expression language is used to express a set of ISA filters in a string expression. The expression language consists of two, very similar, grammars. One grammar is used to express generic digit filters. The other grammar is used to express protocol filters for each probe type. The two grammars can be combined into a single expression, but internally the generic digit expression will be kept separate from each probe type's protocol filter expression. The different expressions can be combined into a single Boolean expression in the following manner:

```
( DIGIT_FILTER_EXPRESSION ) & ( ( G10_PROTOCOL_EXPRESSION ) | ( SPI_PROTOCOL_EXPRESSION ) )
```

General Language Constraints

Both expression languages have the following constraints:

- All expressions must be in Disjunctive Normal Form. Conjunctive expressions will be rejected.
- No nesting of parenthesis is allowed in an expression.

Digit Filter Expression Language

The digit filter expression language is used to define a set of generic digit filters for ISA.

Grammar

The grammar for expressing digit filters is defined as following parsing expression grammar:

```
SPACE <- <One or more spaces>
QUOTE <- '
FILTER_NAME <- QUOTE <Valid Filter Name> QUOTE
FILTER_VALUE <- QUOTE <Comma separated values> QUOTE
OR <- |
AND <- &
MATCHES_ANY <- =
OPEN_PAREN <- (
CLOSE_PAREN <- )
FILTER <- FILTER_NAME SPACE* MATCHES_ANY SPACE* FILTER_VALUE
AND_FILTER <- SPACE* AND SPACE* FILTER
FILTER_EXP <- OPEN_PAREN SPACE* FILTER AND_FILTER* SPACE* CLOSE_PAREN
OR_FLT_EXP <- SPACE* OR SPACE* FILTER_EXP
EXPRESSION <- FILTER_EXP OR_FLT_EXP*
```

Filter Name

The name of a filter should match the name displayed in the trace client. For filters that are specific to a certain probe, the probe type (G10 or SPI) must be appended to the beginning of the name followed by a colon.

Examples

The following are examples of digit filter expressions:

```
( 'IPADDR' = '10.0.0.1' )
( 'IMSI' = '123' & 'MSISDN' = '456, 789' ) | ( 'IPADDR' = '123., 222.' )
('APN' = '123*' & 'IMEI' = '456*') | ('IPADDR' = '255.*') | ('G10:DPC' = '123-123-123')
```

Protocol Filter Expression Language

The protocol filter expression language is used to define a set of protocol filters. ISA will accept one expression per probe type. This means there will be an expression for G10 protocol filters, and a separate expression for spI protocol filters. Each probe specific expression will be effectively OR'd together.

Grammar

The grammar for protocol filters is defined as following parsing expression grammar:

```
SPACE <- <One or more spaces>
QUOTE <- '
COMMA <- ,
OPEN_BRKT <- [
CLOSE_BRKT <- ]
PROTO_NAME <- QUOTE <Valid Protocol Name> QUOTE
PROTO_FIELD <- <Valid Protocol Field Name>
PROTO_FILTER <- QUOTE PROTO_FIELD QUOTE
PROTO_VALUE <- <Field Value> [ OPEN_BRACKET <Field Options> CLOSE_BRACKET ] [ COMMA PROTO_VALUE ]
PROTO_VALUES <- QUOTE PROTO_VALUE QUOTE
OR <- |
AND <- &
INCLUDES_ANY <- =
OPEN_PAREN <- (
CLOSE_PAREN <- )
COLON <- :
FILTER <- PROTO_FILTER SPACE* INCLUDES_ANY SPACE* PROTO_VALUE
AND_FILTER <- SPACE* AND SPACE* FILTER
FILTER_EXP <- PROTO_NAME COLON SPACE* OPEN_PAREN SPACE* FILTER AND_FILTER* SPACE* CLOSE_PAREN
OR_FLT_EXP <- SPACE* OR SPACE* FILTER_EXP
EXPRESSION <- FILTER_EXP OR_FLT_EXP*Field Value Format
```

Field Value Format

Field values may have options as well as sub groups, code groups, and codes.

Options

Options shall be placed at the end of the value inside brackets ([and]). The options should be separated by semicolons (;). Below is an example of a value with options:

```
Bearer Resource Command[Timed Out;Failed]
```

Sub Group, Code Group, and Code

Sub group, code group, and codes must be specified at the beginning of the value. The values must be in correct order (sub group, code group, code). Each value is separated by semicolons (;). If a value is blank, it is omitted from the list, but still separated by semicolons. Therefore, there should always be two semicolons.
The sub group, code group, and code are separated from the rest of the value by a forward slash (/).
Below is an example of a value including a sub group and a code:

```
GTPv2-C;;4/RAT Changed from 3GPP to Non-3GPP
```

Examples

The following are examples of protocol filter expressions:

filterValue

[String] Field to limit search by

Used in conjuction with filterBy as follows:
filterBy ={key}&filterValue ={value}

where key/value pairs are:

Created	session create time
File Name	session name
Comment	session comments
File Size	session size

See parameter details for more detail on each key.

example: filterBy=created & filterValue=2012-10-21T06:52:00.000

flowApplicationFilterExpression

[String] Filters flow applications in a capture session. See filterExpression for details.

example: flowApplicationFilterExpression = AD DSAOP,AD Backup

flowApplications

[List of comma separated Strings] List of flow applications

example: flowApplications = AIM, Vonage

flowMessageInspection

[Boolean] Flag designating whether flow message inspection is enabled or not

values: true|false

flowProtocolFilterExpression

[String] Filters flow protocols in a capture session. See filterExpression for details.

example: flowProtocolFilterEpxression = DHCP, DNS

flowProtocols

[List of comma separated Strings] List of flow protocols

example: flowProtocols = DHCP, DNS

g10ProtocolFilterExpression

[String] Filters g10 protocols in a capture session. See filterExpression for details.

example: g10ProtocolFilterExpression = 'T.38':('Session Type' = 'T.38') | 'GTPv2':('Session Type' = 'Non 3GPP Access, Path Management') | 'GTPv2':('Transaction Type' = 'GTPv2-C;;/Bearer Resource Command[Timed Out;Failed]' & 'Response Code' = 'GTPv2-C;;4/RAT Changed from 3GPP to Non-3GPP')

genericFilterExpression

[String] Filters generic elements in a capture session. See filterExpression for details.

example: genericFilterExpression = ('APN' = '123*' & 'IMEI' = '456*') | ('IPADDR' = '255.*')

gsoftProtocolFilterExpression

[String] Filters gsoft protocols in a capture session. See filterExpression for details.

example: gsoftProtocolFilterExpression = 'T.38':('Session Type' = 'T.38') | 'GTPv2':('Session Type' = 'Non 3GPP Access, Path Management') | 'GTPv2':('Transaction Type' = 'GTPv2-C;;/Bearer Resource Command[Timed Out;Failed]' & 'Response Code' = 'GTPv2-C;;4/RAT Changed from 3GPP to Non-3GPP')

gmtOffset

[String] Difference between local time and GMT.

example: gmtOffset = +0200

groupByRecord

[Boolean] Flag that specifies whether or not to sort messages by record.

values: true|false

groupedNodeTypes

[String] Designation for node types that are grouped in a common cloud.

List separated by a "|"

example: groupedNodeTypes = UNKNOWN

groups

[Comma separated list of Strings] Name of a collection of monitored objects to refine a topology search.

example: groups = nodegroup:1, nodegroup:2

hasAdminRole

[Boolean] Flag that designates whether a user has the admin role or not.

values: true|false

hasVipRole

[Boolean] Flag that designates whether a user has the VIP role or not.

values: true|false

header

[Boolean] Flag designating whether the header is displayed in an exported report.

values: true|false

includedMessages

[Comma separated list of Strings] Messages to be included in an export.

example: includedMessages = g10:123, spi:423

index

1. (Messages): [String] The index of the first message to retrieve.
2. (RSR): [String] Current page index

example: index = 0

isPersistence

[Boolean] Persistence for On Demand Capture, “true” indicates persistence is selected in Client, a persistence profile in plist should be created. Otherwise, no persistence is requested.

values: true|false

keepMpcTaskOnComplete

[Boolean] flag to indicate whether to keep the MPC task when the MPC search is completed. “True” indicates to keep the MPC task even the MPC search is completed, “false” indicates to discard the MPC task when the MPC search is completed to save memory.

values: true|false

ladder

[Boolean] Flag designating whether the ladder is displayed in an exported report.

values: true|false

lastEventTime

Starting point for retrieving objects on the next poll (long value in milliseconds since epoch).

example: lastEventTime = 1346266281696

loginUserId

[String] The identifier for the currently logged in user performing a query

example: loginUserId = admin

mediaCapture

[Boolean] Flag designating whether media capture is enabled or not

values: true|false

mediaTypeFilterExpression

[String] Filters media types in a capture session.

example: mediaTypeFilterExpression = Any RTP, Audio

merge

[Boolean] Whether or not to merge and sort any intermediate export files. This parameter is only used if the parameter withUserPlanePDUs is set to true.

Values: true | false

example: merge = true

messageColumnSetName

[String] identifier for message column set

example: messageColumnSetName = message

messageDetails

[Boolean] Flag designating whether the message details are displayed in an exported report.

values: true|false

messageEndTime

[String] End time for a message, the number string of milliseconds since January 1, 1970, 00:00:00 GMT represented by this date.

example: messageEndTime = 1394502569340

messageId

[String] The identifier for a message

Follows pattern: [g10][spi]:[PDU][FLOW_PDU][FLOW]:<generated number>

examples:

```
messageId = spi:PDU:1
messageId = g10:FLOW_PDU:13
messageId = g10:FLOW:4
```

messageStartTime

[String] Start time for a message, the number string of milliseconds since January 1, 1970, 00:00:00 GMT represented by this date.

example: messageStartTime = 1394502569340

messageTable

[Boolean] Flag designating whether the message table is displayed in an exported report.

values: true|false

messageTableColumns

[List of delimited Strings] List of columns to populate an exported message table.

List separated by “$-$” for CSV and HTML, and a comma for ISA exports.

example: messageTableColumns = Start Time$-$End Time

monitoredObjects

[String] The traffic monitoring device that are queried in a capture

The format depends on the value of monitoredObjectType.

If monitoredObjectType is PROBE, the value of monitoredObjects should be a comma-separated string of probe names.

Example:

monitoredObjects=probe1, probe2, probe3

Iif monitoredObjectType is NODE, the value of monitoredObjects should be a comma-separated string of:

- node names
- colon-separated node pairs
- any combination of node names and colon-separated node pairs

Examples:

monitoredObjects = node1, node2, node3

monitoredObjects = node1:remote1,node2:remote2,node1:remote2

monitoredObjects = node1,node2:remote1,node3,node4:remote2

monitoredObjectsDisplayName

[String] Label used to display monitored object in user interface.

Example: moniotredOjbectsDisplayName = g113 (Probe), Group1 (Probe Group)

monitoredObjectType

[Enum] Particular variation of a monitored object.

values: node|probe

name

[String] User supplied label for a detached session.

Example: name = MyDetachedSession

names

[Comma separated list of Strings] A collection of node names to gather pointcodes for.

example: names = node1, node2

nodes

[String] Network elements that connect two PDUs or Flows

List separated by a "|"

example: nodes = MME/108.188.1.1|SGW/100.1.1.1

nodeOrdering

[String] Order in which nodes are displayed in a report

Comma separated list.

example: nodeOrdering = AAA , UNKNOWN

nodeOrderingEnabled

[Boolean] Flag designating whether node ordering a implemented or not

values: true|false

nodeTypes

[Comma separated list of Strings] Type of a node

example: nodeTypes = AAA, UNKNOWN

options

[Enum] Topology filter to limit probe search

value: includeManagedProbes

overiddenNodes

[String] Nodes whose type has been manually overridden

example: overriddenNodes = PDN-GW/10.10.97.5:AAA

overriddenNodeTypes

[String] Type manually assigned to overridden node

Comma separated list.

example: overriddenNodeTypes = 10.234.59.254=UNKNOWN|AAA

pageSize

[String] The number of messages to retrieve.

example: pageSize = 25

password

[String] The user's login access code.

example: password = changeMe

pinnedNodes

[String] Nodes pinned by user to persist on ladder

List separated by | for HTML exports and comma separated for ISA exports.

example: pinnedNodes = MME/10.10.97.3|PDN-GW/10.10.97.5

pluginType

[Enum] Interface between the sofware and probe.

values: common|g10|spi

pluginTypes

[Comma separated list of Enums] Interface between the sofware and probe.

values: common|g10|spi

example: common, g10, spi

pointcodeFilter

[String] Filters pointcodes in a capture session. See filterExpression for details.

example: pointcodeFilter = Type:OPC, Pointcodes : SIGTRAN_NODE/032-032-000, gop_BSC1

pointcodeFilterExpression

[String] Filters pointcodes in a capture session.

example: pointcodeFilterExpression = Type:OPC, Pointcodes : SIGTRAN_NODE/032-032-000, gop_BSC1

preference

[String] Value of preference to be set.

example: preference="SUBSCRIBER IP Node","Generic - OnDemand","eNodeB","NodeB","BSC"

preferenceName

[String] The identifier of the preference to be retrieved

example: preference Name = ladderNodeOrdering

probeType

[Enum] Type of probe

values: g10|spi

profileName

[String] Identifier for VIP profile

example: profileName = vip1

recordColumnSetName

[String] identifier for record column set

example: recordColumnSetName = record

recordId

[String] Unique identifier for to a record.

example: recordId = mpc:1577166801

recordIdsCsv

[List of comma separated Strings] List of unique record identifiers.

example: recordIdsCsv = mpc:1577166801, mpc:1577324538, mpc:1547363898

recordTable

[Boolean] Flag designating whether the record table is displayed in an exported report.

values: true|false

recordTableColumns

[List of delimted Strings] List of columns to populate an exported record table.

List separated by “$-$” for CSV and HTML, and a comma for ISA exports.

example: recordTableColumns = SessionId$-$IMEISV$-$TransportProtocols$-$ECGI$-$ApplicationID

requestId

[String] Unique identifier for an export.

example: requestId = 1001

renameSessionRecordStr

[String] Identifier used to rename a session record

example: renameSessionRecordStr = newName

securitytoken

[String] A unique hashed identifier tied to an authenticated user.

example:securitytoken = deLN6cBDXd6W2D1fNZxO20DOx98OeCxZ

selectedUserId

[String] The identifier for a user that a query is searching against

example: selectedUserId = user1

sessionSize

[String] A unique hashed identifier tied to an authenticated user.

example:securitytoken = deLN6cBDXd6W2D1fNZxO20DOx98OeCxZ

sorterBy

[String] Database column to sort

example: sorterBy = captureId

sorterOrder

[Enum] Order in which to sort

values: Ascending|Descending|Unsorted

spiProtocolFilterExpression

[String] Filters spi protocols in a capture session. See filterExpression for details.

example: spiProtocolFilterExpression ='GSM AIF':('Response Code' = 'BSSMAP Cause;;1/Radio interface failure') | 'GTP':('Status Event' = 'Gtp Closed due to SGSN Context, Gtp PDP context active, Gtp Procedure completed')

split

[Boolean] Whether or not to split the export file. If this parameter is true, one large export file will be returned, ignoring the values of splitFileSize and splitFileSizeUnits. This parameter is only used if the parameter withUserPlanePDUs is set to true.

Values: true | false

example: split = true

splitByType

[Boolean] Flag indicating whether to split different types of PDUs into separate export files.

The different types of PDUs are:

- Control Plane - All Control Plane packets
- IP Reassembled – User Plane Synthetic packets due to IP reassembly
- L4 Reassembled – User Plane Synthetic packets due to L4 reassembly
- Raw - User Plane packets not requiring any reassembly

Values: true|false

example: splitByType = true

splitFileSize

[Integer] The maximum size (in the units specified by splitFileSizeUnits) of an individual export file.

This parameter is only used when the parameter withUserPlanePDUs is specified.

Values: 1 - 2,147,483,647

example: splitFileSize = 2048

splitFileSizeUnits

[String] The unit to apply to splitFileSize.

This parameter is only used when the parameter withUserPlanePDUs is specified.

Values: MB|GB

example: splitFileSizeUnits = GB

startTime

[String] The beginning time of a query.

Follows ISO-8061 format: yyyy-mm-ddTHH:mm:ss.SSS

example: startTime = 2012-10-21T06:52:00.000

statusMode

[Enum] CLIENT mode is used to display high level information; DEBUG mode is used to display infromation from all levels.

values: CLIENT|DEBUG

stopAction

[Enum] Action to take on message details stop

values: STOP|COLLAPSE

STOP - Stop any outstanding commands for Flow Details do not collapse the flow

COLLAPSE - Collapse the flow

example: stopAction = STOP

timeTarget

[Enum] Define the active time to be based on ACTIVITY time or MONITORING time, default value is ACTIVITY.

values: ACTIVITY|MONITORING

example: timeTarget = ACTIVITY

topologyType

[Enum] Specify the topology type to be queried.

values: spi|g10|gsoft|all

example: topologyType = all

useDefault

[Boolean] Flag designating wheter to override user preferences with default value or not

Values: true|false

example: useDefault = true

username

[String] The user's unique identifier.

example: username = admin

userroles

[Comma separated list of Strings] Key to allocate permissions to licensed features

example: userroles = ROLE_3rd_PARTY_API, ROLE_ISA

withUserPlanePDUs

[Boolean] Flag to indicate whether to attempt to retrieve all user plane PDUs for flow summaries present in the exported records.

If this parameter is specified, the following parameters must also be specified:

- fileSize
- fileSizeUnits
- splitFileSize
- splitFileSizeUnits

Values: true|false

example: withUserPlanePDUs = true

workflowName

[String] Name of the workflow

example: workflowName = Session Trace

3.2 MAPPING QUICK REFERENCE

3.2.0 User Authentication

3.2.0.1 Login

POST: {baseURL}/login
params (req):password, username

3.2.0.2 Logout

POST: {baseURL}/logout
headers (req): securitytoken

3.2.0.3 Heartbeat

POST: {baseURL}/hearbeat
headers (req): securitytoken

3.2.1 Application Information

3.2.1.1 Get Application Information

GET: {baseURL}/appinfo/{pluginType}
path vars (req): pluginType (spi only)

3.2.2 Capture

3.2.2.1 Get Captures

GET: {baseURL}/capture/captureSession
headers (req): username
headers (opt): userroles

3.2.2.2 Start Capture

POST: {baseURL}/capture
headers (opt): username, userroles
params (req): monitoredObjects, monitoredObjectType
params (opt): endTime, flowApplicationFilterExpression, flowMessageInspection, flowProtocolFilterExpression, g10ProtocolFilterExpression, genericFilterExpression, mediaCapture, mediaTypeFilterExpression, pointcodeFilterExpression, searchTarget, timeTarget, spiProtocolFilterExpression,startTime,workFlowName,pluginType,isPersistence, timeTarget,gsoftProtocolFilterExpression,keepMpcTaskOnComplete, workflowName, monitoredObjectsDisplayName

3.2.2.3 Stop Capture

PUT:

- {baseURL}/capture (When nginx is turned OFF)
- {baseURL}/InstanceId/capture (When nginx is turned ON)

headers (req): captureId

3.2.2.4 Close Capture

DELETE:

- {baseURL}/capture (When nginx is turned OFF)
- {baseURL}/InstanceId/capture (When nginx is turned ON)

headers (req): captureId

3.2.2.5 Resume Capture

PUT:

- {baseURL}/capture/resume (When nginx is turned OFF)
- {baseURL}/InstanceId/capture/resume (When nginx is turned ON)

headers (req): captureId

3.2.2.6 Attach Capture

PUT:

- {baseURL}/capture/attach (When nginx is turned OFF)
- {baseURL}/InstanceId/capture/attach (When nginx is turned ON)

headers (req): captureId

3.2.2.7 Detach Session

POST:

- {baseURL}/capture/detach (When nginx is turned OFF)
- {baseURL}/InstanceId/capture/detach (When nginx is turned ON)

headers (req): captureId, username

params (opt): duration, name

3.2.2.8 Terminate Sessions

DELETE:

- {baseURL}/capture/terminate (When nginx is turned OFF)
- {baseURL}/InstanceId/capture/terminate (When nginx is turned ON)

headers (req): username

params (req): captureId

3.2.2.9 Touch Capture

PUT:

- {baseURL}/capture/touch (When nginx is turned OFF)
- {baseURL}/InstanceId/capture/touch (When nginx is turned ON)

headers (req): captureId

3.2.2.10 Set Post Capture Filter Session

POST:

- {baseURL}/capture/post (When nginx is turned OFF)
- {baseURL}/InstanceId/capture/post (When nginx is turned ON)

headers (req): captureId

params (opt): filterExpression, timezone

3.2.2.11 Detach Enable

POST: {baseURL}/capture/detachEnable

3.2.2.12 Get Chunk Records

GET:

- {baseURL}/capture/records/{chunkSize} (When nginx is turned OFF)
- {baseURL}/InstanceId/capture/records/{chunkSize} (When nginx is turned ON)

headers (req): captureId

path vars (req): chunkSize

3.2.2.13 Get Records

GET:

- {baseURL}/capture/records (When nginx is turned OFF)
- {baseURL}/InstanceId/capture/records (When nginx is turned ON)

headers (req): captureId

3.2.2.14 Monitored Elements

GET:

- {baseURL}/capture/monitoredElements (When nginx is turned OFF)
- {baseURL}/InstanceId/capture/monitoredElements (When nginx is turned ON)

headers (req): captureId

3.2.3 Events

3.2.3.1 Get Events without Time

GET:

- {baseURL}/event (When nginx is turned OFF)
- {baseURL}/InstanceId/event (When nginx is turned ON)

headers (req): captureId

3.2.3.2 Get Events with Time

GET:

{baseURL}/event/{lastEventTime} (When nginx is turned OFF)
{baseURL}/InstanceId/event/{lastEventTime} (When nginx is turned ON)

headers (req): captureId

path vars (req): lastEventTime

3.2.4 Export

3.2.4.1 Start Export

POST:

- {baseURL}/export/doExport/{exportType}/{recordIdsCsv} (When nginx is turned OFF)
- {baseURL}/InstanceId/export/doExport/{exportType}/{recordIdsCsv} (When nginx is turned ON)

headers (req): captureId

path vars (req): exportType, recordIdsCsv

params (req): requestId

params (opt): see mapping

3.2.4.2 Export Now

PUT:

- {baseURL}/export/exportNow/{requestId} (When nginx is turned OFF)
- {baseURL}/InstanceId/export/exportNow/{requestId} (When nginx is turned ON)

headers (req): captureId

path vars (req): requestId

3.2.4.3 Cancel Now

PUT:

- {baseURL}/export/cancelNow/{requestId} (When nginx is turned OFF)
- {baseURL}/InstanceId/export/cancelNow/{requestId} (When nginx is turned ON)

headers (req): captureId

path vars (req): requestId

params (opt): doExportToRepository, exportTime, fileName, requestId, username

3.2.4.4 Get Plist Properties

GET: {baseURL}/export/getPlistProperties

3.2.5 Filters

3.2.5.1 Get Filter Elements

GET: {baseURL}/filter/{pluginTypes}
headers (opt): username, userroles
path vars (req): pluginTypes

3.2.5.2 Get Applications

GET: {baseURL}/filter/application/{pluginType}

path vars (req): pluginTypes

3.2.5.3 Get Advanced Options

GET: {baseURL}/filter/advanced/{pluginTypes}
headers (opt): userroles
path vars (req): pluginTypes

3.2.5.4 Get Flow Applications

GET: {baseURL}/filter/advanced/flowApplications

3.2.6 Filter Validation

3.2.6.1 ValidateFilters

POST: {baseURL}/filterExpression/validate
params (req): monitoredObjects
params (opt): monitoredObjectType, genericFilterExpression, g10ProtocolFilterExpression, spiProtocolFilterExpression, pointcodeFilterExpression, mediaTypeFilterExpression, flowProtocols, flowApplications

3.2.6.2 Validate Proto Filter

POST: {baseURL}/filter/validation/proto
params (req): expression, probeType

3.2.6.3 Validate Digit Filter

POST: {baseURL}/filter/validation/digit
params (req): expression, pluginTypes

3.2.6.4 Validate Flow Applications

POST: {baseURL}/filter/validation/flowApplications
params (req): flowApplications

3.2.6.5 Validate Flow Protocols

POST: {baseURL}/filter/validation/flowProtocols
params (opt): flowProtocols

3.2.7 Messages

3.2.7.1 Start Messages

POST:

- {baseURL}/capture/{recordId} (When nginx is turned OFF)
- {baseURL}/InstanceId/capture/{recordId} (When nginx is turned ON)

headers (req): captureId

path vars (req): recordId

3.2.7.2 Get Messages

GET:

- {baseURL}/capture/{recordId} (When nginx is turned OFF)
- {baseURL}/InstanceId/capture/{recordId} (When nginx is turned ON)

headers (req): captureId

path vars (req): recordId

params (opt): index, pageSize

3.2.7.3 Stop Messages

DELETE:

- {baseURL}/capture/{recordId} (When nginx is turned OFF)
- {baseURL}/InstanceId/capture/{recordId} (When nginx is turned ON)

headers (req): captureId

path vars (req): recordId

3.2.8 Message Details

3.2.8.1 Start Message Details

POST:

- {baseURL}/capture/{recordId}/{messageId} (When nginx is turned OFF)
- {baseURL}/InstanceId/capture/{recordId}/{messageId} (When nginx is turned ON)

headers (req): captureId

path vars (req): messageId, recordId

3.2.8.2 Stop Message Details

DELETE:

- {baseURL}/capture/{recordId}/{messageId} (When nginx is turned OFF)
- {baseURL}/InstanceId/capture/{recordId}/{messageId} (When nginx is turned ON)

headers (opt): captureId

path vars (req): recordId, messageId

params (req): stopAction

3.2.8.3 Get Message Details Status

PUT:

- {baseURL}/capture/{recordId}/{messageId} (When nginx is turned OFF)
- {baseURL}/InstanceId/capture/{recordId}/{messageId} (When nginx is turned ON)

headers (opt): captureId

path vars (req): messageId, recordId

3.2.9 MPC

3.2.9.1 Merge

POST:

- {baseURL}/mpc/split (When nginx is turned OFF)
- {baseURL}/InstanceId/mpc/split (When nginx is turned ON)

headers (req): captureId
params (req): recordId

3.2.9.2 Reset

POST:

- {baseURL}/mpc/reset (When nginx is turned OFF)
- {baseURL}/InstanceId/mpc/reset (When nginx is turned ON)

headers (req): captureId
params (req): recordId

3.2.9.3 Split

POST:

- {baseURL}/mpc/split (When nginx is turned OFF)
- {baseURL}/InstanceId/mpc/split (When nginx is turned ON)

headers (req): captureId
params (req): recordId

3.2.9.4 Retrieve

POST:

- {baseURL}/mpc/retrieve (When nginx is turned OFF)
- {baseURL}/InstanceId/mpc/retrieve (When nginx is turned ON)

headers (req): captureId
params (req): recordId

3.2.9.5 Process Limited

POST:

- {baseURL}/mpc/process_limited (When nginx is turned OFF)
- {baseURL}/InstanceId/mpc/process_limited (When nginx is turned ON)

headers (req): captureId
params (req): recordId

3.2.9.6 Process Full

POST:

- {baseURL}/mpc/process_full (When nginx is turned OFF)
- {baseURL}/InstanceId/mpc/process_full (When nginx is turned ON)

headers (req): captureId

params (req): recordId

3.2.9.7 Stop

POST:

- {baseURL}/mpc/stop (When nginx is turned OFF)
- {baseURL}/InstanceId/mpc/stop (When nginx is turned ON)

headers (req): captureId

3.2.10 Preference

3.2.10.1 Get User Preferences

GET: {baseURL}/preference/{preference Name}
headers (req): username
path vars (req): preferenceName
params (opt): useDefaults

3.2.10.2 Set User Preferences

POST: {baseURL}/preference/{preference Name}
headers (req): username
path vars (req): preferenceName
params (req): preference

3.2.11 Status

3.2.11.1 Get Capture Status

GET:

- {baseURL}/status/getCaptureStatus (When nginx is turned OFF)
- {baseURL}/InstanceId/status/getCaptureStatus (When nginx is turned ON)

headers (req): captureId

3.2.11.2 Get Message Status

GET:

- {baseURL}/status/getMessageStatus/{recordId} (When nginx is turned OFF)
- {baseURL}/InstanceId/status/getMessageStatus/{recordId} (When nginx is turned ON)

headers (req): captureId

path vars (req): recordId

3.2.11.3 Get Status Task

GET:

- {baseURL}/status/getCaptureStatus (When nginx is turned OFF)
- {baseURL}/InstanceId/status/getCaptureStatus (When nginx is turned ON)

headers (req): captureId

3.2.12 Topology

3.2.12.1 Get Probes

GET: {baseURL}/topology/probes
params (opt): groups, options, pluginType

3.2.12.2 Get Monitored Nodes

GET: {baseURL}/topology/nodes
params (opt): groups, pluginType

3.2.12.3 Get Monitored Links

POST: {baseURL}/topology/links
params (opt): pluginType, pluginIds

3.2.12.4 Get Monitored G10 Nodes

GET: {baseURL}/topology/g10/nodes
params (opt): groups

3.2.12.5 Get Monitored SPI Nodes

GET: {baseURL}/topology/spi/nodes
params (opt): groups

3.2.12.6 Get Monitored Node Groups

GET: {baseURL}/topology/nodeGroups
params (opt): pluginType

3.2.12.7 Get Monitored SPI Node Groups

GET: {baseURL}/topology/spi/nodeGroups

3.2.12.8 Get Monitored G10 Node Groups

GET: {baseURL}/topology/g10/nodeGroups

3.2.12.9 Get Point Codes

GET: {baseURL}/topology/pointCodes
params (opt): names

3.2.12.10 Get Probe Groups

GET: {baseURL}/topology/probeGroups

3.2.12.11 Get Monitored Objects Groups

GET: {baseURL}/topology/groups

3.2.12.12 Get Node Types

GET: {baseURL}/topology/nodeTypes

3.2.12.13 Get Links

GET: {baseURL}/topology/links

3.2.13 Remote Server Repository

3.2.13.1 Get Record Count by User

GET: {baseURL}/repository/getRecordCountByUser
params (req): selectedUserId , loginUserId, hasAdminRole
params (opt): count, filterBy, filterValue, hasAdminRole, index, loginUserId, selectedUserId, sorterBy, sorterOrder

3.2.13.2 Get Record Count by Profile

GET: {baseURL}/repository/getRecordCountByProfile
params (req): profileName, loginUserId, hasVipRole
params (opt): count, filterBy, filterValue, hasVipRole, index, loginUserId, profileName, sorterBy, sorterOrder

3.2.13.3 Get Capture List by Ids

GET:

- {baseURL}/repository/getCaptureListbyIds (When nginx is turned OFF)
- {baseURL}/InstanceId/repository/getCaptureListbyIds (When nginx is turned ON)

params (req): captureId

3.2.13.4 Get Capture List by User

GET: {baseURL}/repository/getCaptureListByUser
params (req): count, hasAdminRole, index, loginUserId, selectedUserId

3.2.13.5 Get Capture List by Profile

GET: {baseURL}/repository/getCaptureListByProfile
params (req): count, filterBy, filterValue, hasAdminRole, index, loginUserId, selectedUserId, sorterBy, sorterOrder

3.2.13.6 Delete Capture Record

DELETE:

- {baseURL}/repository/deleteCaptureRecord (When nginx is turned OFF)
- {baseURL}/InstanceId/repository/deleteCaptureRecord (When nginx is turned ON)

params (req): captureId

3.2.13.7 Toggle Capture Record

POST:

- {baseURL}/repository/toggleCaptureRecord (When nginx is turned OFF)
- {baseURL}/InstanceId/repository/toggleCaptureRecord (When nginx is turned ON)

params (req): captureId

3.2.13.8 Rename Capture Record

POST:

- {baseURL}/repository/renameCaptureRecord (When nginx is turned OFF)
- {baseURL}/InstanceId/repository/renameCaptureRecord (When nginx is turned ON)

params (req): captureId, renameSessionRecordStr

3.2.13.9 Download Capture Session

POST:

- {baseURL}/repository/downloadCaptureSession (When nginx is turned OFF)
- {baseURL}/InstanceId/repository/downloadCaptureSession (When nginx is turned ON)

params (req): captureId

3.2.13.10 Get User List

GET: {baseURL}/repository/getUserList
params (req): hasAdminRole

3.2.13.11 Get Profile List

GET: {baseURL}/repository/getProfileList
params (req): hasVipRole

3.3 SOME GENERIC 3RD PARTY API WORKFLOWS

3.3.0 Html Export Workflow

Figure 1 shows the HTML Export workflow.

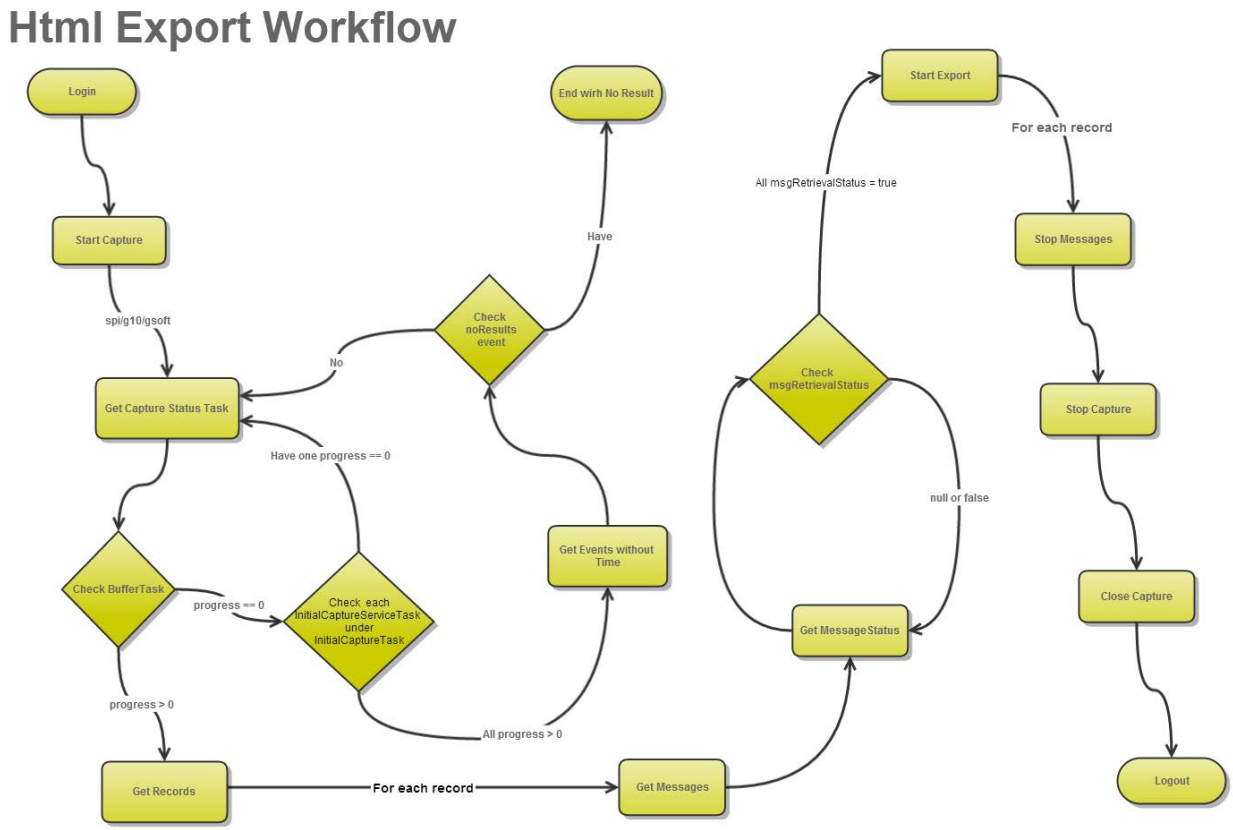


Figure 1 HTML Export Workflow

3.3.1 Pcap Export Workflow

Figure 1 shows the Pcap Export workflow.

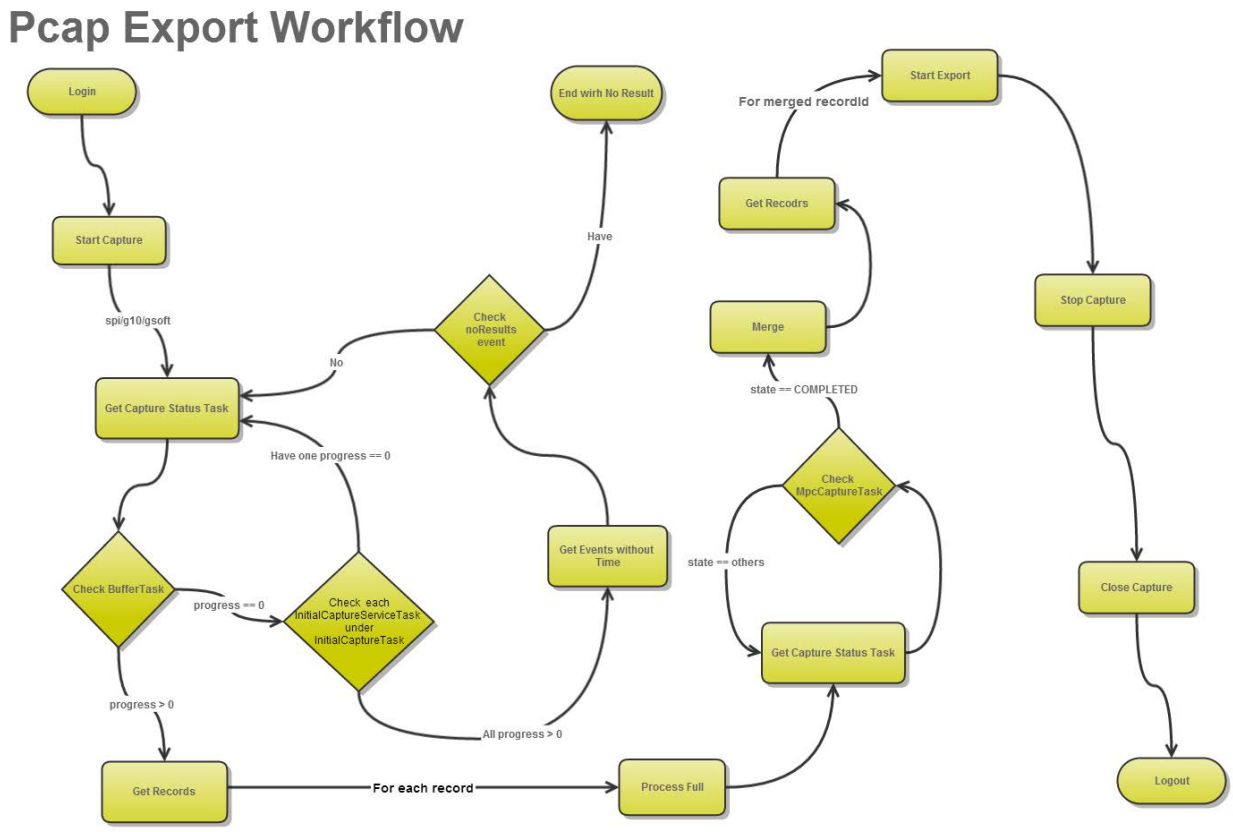


Figure 2 Pcap Export Workflow