



**Communication Gateway
Developer Reference**

1. Contents

2. Notice	4
3. Component Overview	5
Web service (REST based)	5
ActiveX Control	5
Microsoft .Net DLL	5
4. Web service (REST based)	6
Authorisation	6
Endpoint Resources	7
/endpoint/{number}	7
/endpoint/instances.....	7
/endpoint/{number}/recordingstate?state={state}.....	8
/endpoint/{number}/tag?field={field}&value={value}&append={append}.....	9
/endpoint/{number}/timeline?value={value}.....	10
Call Resources	12
/call/{recid}	12
/call?logicalcall={logicalcallid}.....	12
/call/{recid}/recording?format={format}.....	14
/call/recording?logicalcall={logicalcallid}&segments={segments}&format={format}.....	14
User Resources.....	16
/users?username={username}.....	16
Query Resources	17
/query.....	17
Compliance Resources	20
/endpoint/{number}/compliancemonitor/{instance}/add?timeout={timeout}.....	20
/endpoint/{number}/compliancemonitor/{instance}/remove.....	20
/compliance/monitors	21
/compliance/configuration	21
/endpoint/compliance/references	22
Utility Resources	24
/utility/newguid	24
5. ActiveX Control	25
Requirements.....	25
Connection Details.....	25
Properties.....	25
string CommunicationGateway { get; set;}.....	25

string ApplicationID { get; set;}	25
Methods	26
string Endpoint_Get(string Number)	26
void Endpoint_SetRecordingState (string Number, string State)	26
void Endpoint_Tag(string Number, int Field, string Value)	26
void Endpoint_SetCustomTimeline(string Number, string Value).....	26
6. Microsoft .Net DLL	27
Requirements.....	27
Connection Details	27
Properties.....	27
string CommunicationGateway { get; set;}	27
string ApplicationID { get; set;}	27
Methods	28
string Endpoint_Get(string Number)	28
void Endpoint_SetRecordingState (string Number, string State)	28
void Endpoint_Tag(string Number, int Field, string Value)	28
7. Result Codes.....	29
Code	29
Reason.....	29
8. URL Formats.....	30

2. Notice

Xarios reserves the right to make changes or improvements to the product and/or information described in this document at any time and without notice. Every attempt has been made to ensure the information is accurate and complete; however, Xarios is not responsible for any inaccuracies or omissions in this or any of its other publications. In no event is Xarios liable for any incidental, consequential, or indirect damages (including but not limited to loss of business profits, business interruption, or loss of information) arising out of the use or inability to use the information in this document and/or the products described herein. This includes any claim by any other party.

Developers assume full responsibility when using the Communication Gateway API. Xarios, therefore, is not liable for any problems caused by applications developed using this product.

3. Component Overview

There are 3 different options available for integration into the Communication Gateway, as shown in Table 3.1. Depending on the environment that is been used to develop the application, the most appropriate option can be selected.

Component	Description
Web service (REST based)	RESTful based web service
ActiveX Control	Microsoft Windows ActiveX component
Microsoft .Net DLL	Microsoft .Net Framework (v4.0) Assembly

Table 3.1

The REST API is the underlying interface for all the API components. This is the most direct way for accessing the API. All the other components are wrappers round the web service. Features within the web service will generally be available within the other component unless stated otherwise.

Web service (REST based)

The Web service provides a REST (Representation State Transfer) style API that can be consumed over a range of different platforms, i.e. Windows, Linux, OS X etc. This makes it lightweight in terms of the amount of information (XML) sent back and forth. The integration work involved is relatively easy to do as no specific development tools are required.

ActiveX Control

This is a Microsoft ActiveX COM component that can be used within any environment that supports COM components. This is only supported for use on Microsoft Windows platforms.

Microsoft .Net DLL

This is a Microsoft .Net Framework v4.0 DLL assembly that can be used within Visual Studio .Net environments, for example Microsoft Visual Basic.Net and Microsoft C#. This is only supported for use on Microsoft Windows platforms

4. Web service (REST based)

Authorisation

To be able to use any of the API components the application needs to be authorised. This is handled in the form of a custom HTTP header, *AppID*, which contains a specific Application ID. This is a GUID value that must be pre-configured onto the Communication Gateway.

If any requests are made without a valid Application ID then a 403 Forbidden HTTP status code will be returned.

Endpoint Resources

<code>/endpoint/{number}</code>	
Description	Retrieves information about the calls at the endpoint.
URL Structure	<code>http://{gateway}/endpoint/{number}</code>
Method	GET
Parameters	<i>number</i> : The endpoint device number
Returns	Endpoint information
Example	<code>http://mygateway.local/endpoint/1803</code>
Return XML Sample	<pre><EndpointInfo> <Number>1803</Number> <CallDetails> <CallInfo> <RecorderServer>xariosrecorder</RecorderServer> <RecID>45329</RecID> <RecordingState>Recording</RecordingState> <Serial>22102012152453347</Serial> <Trunk>94107</Trunk> <CallID>2@X407</CallID> <Agent>1021</Agent> <DDI/> <DNIS/> <CLI>0800001234</CLI> <Direction>1</Direction> <HuntGroup/> <AccountCode/> <Field1/> <Field2/> <Field3/> <Field4/> <Field5/> </CallInfo> </CallDetails> </EndpointInfo></pre>
Return Definitions	<p><i>RecorderServer</i>: The call recording server name that this call is being recorded on.</p> <p><i>RecID</i>: The unique call recording identifier for this call on this call recording server. If multiple recording servers are connected then this identifier will not be unique across servers.</p>
Exceptions	404 Not Found: If the endpoint number is invalid or does not exist.

<code>/endpoint/instances</code>	
Description	Retrieves the details of each endpoint instance currently connected to the gateway.

URL Structure	<code>http://{gateway}/endpoint/instances</code>
Method	GET
Parameters	<i>none</i>
Returns	Endpoint instance information for each separate connection.
Example	<code>http://mygateway.local/endpoint/instances</code>
Return XML Sample	<pre> <ArrayOfEndpointInstance> <EndpointInstance> <EndpointNumber>1812</EndpointNumber> <InstanceID>383504</InstanceID> <IPAddress>172.19.20.78</IPAddress> <Port>59348</Port> <UserAgent> Xarios Communicator for MSIE; v999.0.0.1001 r999.0.2365.1 </UserAgent> <Timestamp>2013-02-01T11:06:55.3634471+00:00</Timestamp> <Timeout>15000</Timeout> </EndpointInstance> <EndpointInstance> <EndpointNumber>1803</EndpointNumber> <InstanceID>495ae3e4-00ef-4d58-b037- a39ef75877dd</InstanceID> <IPAddress>172.19.20.88</IPAddress> <Port>65097</Port> <UserAgent> Mozilla/5.0 (Windows NT 6.1; rv:18.0) Gecko/20100101 Firefox/18.0 </UserAgent> <Timestamp>2013-02-01T11:06:58.0675755+00:00</Timestamp> <Timeout>15000</Timeout> </EndpointInstance> </ArrayOfEndpointInstance> </pre>
Return Definitions	<p><i>EndpointNumber</i>: The endpoint number.</p> <p><i>InstanceID</i>: The unique instance ID for this connection.</p> <p><i>IPAddress</i>: The IP address the client is connecting from.</p> <p><i>Port</i>: The TCP port the client is connecting from.</p> <p><i>UserAgent</i>: The User Agent string that determines the type of client.</p> <p><i>Timestamp</i>: The date and time when the client last connected.</p> <p><i>Timeout</i>: The amount of time in seconds as to when this connection will be closed unless the client makes any connection.</p>

`/endpoint/{number}/recordingstate?state={state}`

Description	Sets the call recording state for the active call at the endpoint. If multiple call recorders are recording this call then the state of each call recording will be set.
URL Structure	<code>http://{gateway}/endpoint/{number}/recordingstate?state{state}</code>

Method	GET
Parameters	<i>number</i> : The endpoint device number. <i>state</i> : The call recording state to set for this call.
Parameter Options	<i>state</i> : <i>recording</i> – sets the call to be recorded. Only applies when the call is active and is in the <i>paused</i> state. <i>paused</i> – sets the call recording to be <i>paused</i> . Only applies when the call is active and is in the <i>recording</i> state.
Example	http://mygateway.local/endpoint/1803/recordingstate?state=paused
Return XML Sample	<pre><ArrayOfOperationResult> <OperationResult> <ServerName>xariosrecorder\callrecorder</ServerName> <Code>0</Code> <Reason>Pause request sent</Reason> </OperationResult> </ArrayOfOperationResult></pre>
Exceptions	404 Not Found: If the endpoint number is invalid or does not exist. 405 Method Not Allowed: If the recording state value is not valid.
Codes	1001 Generic error 1002 Endpoint not found 1003 Unable to change recording state 1103 No call in progress 1104 Invalid recording state

/endpoint/{number}/tag?field={field}&value={value}&append={append}

Description	Tags the current active call at the endpoint. The custom tag field number will then be set to the value provided.
URL Structure	http://{gateway}/endpoint/{number}/tag?field={field}&value={value}&append={append}
Method	GET
Parameters	<i>number</i> : The endpoint device number. <i>field</i> : The custom tag field number to update. <i>value</i> : The value to update the custom tag field with. <i>append</i> : Sets if the value should be appended to the current field value.
Parameter Options	<i>field</i> : This sets the field number to update and needs to be a value between 1 and 5. <i>value</i> : This is the value to set the field to. <i>append</i> :

	True or false value to indicate if the value should be appended to the existing field value.								
Example	<code>http://mygateway.local/endpoint/1803/tag?field=1&value=mycustomerref&append=true</code>								
Return XML Sample	<pre><ArrayOfOperationResult> <OperationResult> <ServerName>xariosrecorder\callrecorder</ServerName> <Code>1103</Code> <Reason>No call in progress</Reason> </OperationResult> </ArrayOfOperationResult></pre>								
Exceptions	404 Not Found: If the endpoint number is invalid or does not exist.								
Codes	<table border="0"> <tr> <td>1001</td> <td>Generic error</td> </tr> <tr> <td>1002</td> <td>Endpoint not found</td> </tr> <tr> <td>1102</td> <td>Invalid field number</td> </tr> <tr> <td>1103</td> <td>No call in progress</td> </tr> </table>	1001	Generic error	1002	Endpoint not found	1102	Invalid field number	1103	No call in progress
1001	Generic error								
1002	Endpoint not found								
1102	Invalid field number								
1103	No call in progress								

`/endpoint/{number}/timeline?value={value}`

Description	Adds a custom timeline entry to the current active call at the endpoint. The custom timeline entry will then be shown on the playback timeline.						
URL Structure	<code>http://{gateway}/endpoint/{number}/timeline?value={value}</code>						
Method	GET						
Parameters	<p><i>number</i>: The endpoint device number.</p> <p><i>value</i>: The value to update the custom tag field with.</p>						
Parameter Options	<p><i>value</i>:</p> <p>This is the value to set the custom timeline to.</p>						
Example	<code>http://mygateway.local/endpoint/1803/timeline?value=mucustomtimeline</code>						
Return XML Sample	<pre><ArrayOfOperationResult> <OperationResult> <ServerName>xariosrecorder\callrecorder</ServerName> <Code>1103</Code> <Reason>No call in progress</Reason> </OperationResult> </ArrayOfOperationResult></pre>						
Exceptions	404 Not Found: If the endpoint number is invalid or does not exist.						
Codes	<table border="0"> <tr> <td>1001</td> <td>Generic error</td> </tr> <tr> <td>1002</td> <td>Endpoint not found</td> </tr> <tr> <td>1103</td> <td>No call in progress</td> </tr> </table>	1001	Generic error	1002	Endpoint not found	1103	No call in progress
1001	Generic error						
1002	Endpoint not found						
1103	No call in progress						

Call Resources

/call/{recid}	
Description	Retrieves the details of the call record segment referenced by the RecID.
Min Version	v2.2
URL Structure	http://{gateway}/call/{recid}
Method	GET
Parameters	<i>recid</i> : The identifier for this call segment.
Parameter Options	<i>recid</i> : Each recorded segment of a call is referenced by a unique reference called the recid. This is unique within a site.
Example	http://mygateway.local/call/123456
Return XML Sample	<pre> <callrecord> <logicalcall>false</logicalcall> <attribute> <Key> <callrecordattributetype>recid</callrecordattributetype> </Key> <Value><string>123456</string></Value> </attribute> </callrecord> </pre>
Return Definitions	<p><i>CallRecord</i>: The call record details.</p> <p><i>LogicalCall</i>: If set this this record is the logical call record.</p> <p><i>Attribute</i>: This contains key/value pair information relating to the call. The number of key/value pairs may vary based upon configuration.</p>
Exceptions	<p>404 Not Found: If the recid does not exists.</p> <p>406 Not Acceptable: If the recid is not valid.</p>

/call?logicalcall={logicalcallid}	
Description	Retrieves the details of the logical call, including all call record segments.
Min Version	v2.2
URL Structure	http://{gateway}/call?logicalcall={logicalcallid}
Method	GET

Parameters	<i>logicalcallid</i> : The identifier for this call segment.
Parameter Options	<i>logicalcallid</i> : Each call recorded can be made from several separate segments, the <i>logicalcallid</i> relates all the segments together.
Example	http://mygateway.local/call?logicalcall=12345
Return XML Sample	<pre> <callrecordcollection count="2"> <callrecord> <logicalcall>true</logicalcall> <callsegment> <segmentnumber>1</segmentnumber> <recid>123456</recid> </callsegment> <callsegment> <segmentnumber>2</segmentnumber> <recid>123457</recid> </callsegment> <attribute> <Key> <callrecordattributetype>logicalcallid</callrecordattributetype> </Key> <Value><string>lcd1001</string></Value> </attribute> </callrecord> <callrecord> <logicalcall>>false</logicalcall> <attribute> <Key> <callrecordattributetype>recid</callrecordattributetype> </Key> <Value><string>123456</string></Value> </attribute> </callrecord> <callrecord> <logicalcall>>false</logicalcall> <attribute> <Key> <callrecordattributetype>recid</callrecordattributetype> </Key> <Value><string>123457</string></Value> </attribute> </callrecord> </callrecordcollection> </pre>
Return Definitions	<p><i>CallRecordCollection</i>: This is the collection of different call records that the <i>logicalcall</i> is made from. The call record that has the <i>LogicalCall</i> property set is a summary record that aggregates some of the information together.</p> <p><i>CallSegment</i>: This contains the segment information that shows the specific call record that makes up each segment.</p> <p><i>CallRecord</i>: The call record details.</p> <p><i>LogicalCall</i>: If set this this record is the logical call record.</p> <p><i>Attribute</i>: This contains key/value pair information relating to the call. The number of key/value pairs may vary based upon configuration.</p>
Exceptions	404 Not Found: If the <i>logicalcallid</i> does not exist.

406 Not Acceptable: If the logicalcallid is not valid.

/call/{recid}/recording?format={format}

Description	Retrieves the audio recording file for the call record segment referenced by the RecID.
Min Version	v2.2
URL Structure	http://{gateway}/call/{recid}/recording?format={format}
Method	GET
Parameters	<i>recid</i> : The identifier for this call segment. <i>format</i> : The format to return the audio file
Parameter Options	<i>recid</i> : Each recorded segment of a call is referenced by a unique reference called the recid. This is unique within a site. <i>format</i> : The format can either be WAV or MP3, if none is provided then WAV is returned. MP3: MPEG3, 8kHz, 56kbs, mono, VBR WAV: GSM 6.10, 8kHz, mono
Example	http://mygateway.local/call/123456/recording?format=wav
Returns	This returns the audio file in the requested format.
Exceptions	404 Not Found: If the recid does not exist. 406 Not Acceptable: If the recid is not valid.

/call/recording?logicalcall={logicalcallid}&segments={segments}&format={format}

Description	Retrieves the audio recording file for logical call, including the specified call record segments.
Min Version	v2.2
URL Structure	http://{gateway}/call/recording?logicalcall={logicalcallid}&segments={segments}&format={format}
Method	GET

Parameters	<p><i>logicalcall</i>: The identifier for this call segment.</p> <p><i>format</i>: The format to return the audio file.</p> <p><i>segments</i>: A comma separated list of segment numbers.</p>
Parameter Options	<p><i>format</i>:</p> <p>The format can either be WAV or MP3, if none is provided then WAV is returned.</p> <p>MP3: MPEG3, 8kHz, 56kbs, mono, VBR</p> <p>WAV: GSM 6.10, 8kHz, mono - Not currently supported</p> <p><i>segments</i>:</p> <p>This is a comma separated list of segment numbers to include in the combined audio file. If not specified then all segments will be included.</p>
Example	<p>http://mygateway.local/call?logicalcall=123456/recording?format=mp3</p>
Returns	<p>This returns a combined audio file with all of the segments in the requested format.</p>
Exceptions	<p>404 Not Found: If the logicalcallid does not exist.</p> <p>406 Not Acceptable: If the logicalcallid is not valid.</p>

<code>/users?username={username}</code>	
Description	This returns the details of the user with the given username, domain username or email address configured against their account.
Min Version	v3.0
URL Structure	<code>http://{gateway}/users?username={username}</code>
Method	GET
Parameters	<i>username</i> : The username, domain username or email address of the user.
Example	<code>http://mygateway.local/users?username=domain\user</code>
Return XML Sample	<pre> <userinfo> <userid>0f688ae3-26c3-46fa-838e-0bb210f8ca20</userid> <forename>John</forename> <surname>Smith</surname> <username></username> <domainusername>domain\john</domainusername> <endpoints count="2"> <endpoint>1001</endpoint> <endpoint>1002</endpoint> </endpoints> </userinfo> </pre>
Return Definitions	<p><i>UserID</i>: The unique GUID to identify this user.</p> <p><i>Forename</i>: The forename configured against this user.</p> <p><i>Surname</i>: The surname configured against this user.</p> <p><i>Username</i>: The username configured against this user.</p> <p><i>DomainUsername</i>: The domain username configured against this user.</p> <p><i>Endpoints</i>: The list of endpoint numbers that are assigned to this user.</p>
Exceptions	404 Not Found: If the username is invalid or does not exist.

/query	
Description	This queries the call list to find any matching calls that meet the conditions provided.
Min Version	v3.0
URL Structure	http://{gateway}/query
Method	POST
Parameters	<i>queryExpression</i> : The query expression that details the conditions to use.
Parameter Details	<p><i>queryExpression</i>: This describes the conditions that are to be used for searching for any matching calls. The QueryExpression contains FilterExpressions and these contains a selection of ConditionExpressions. The ConditionExpressions then contain the specific attribute and values to use.</p> <p>The <i>starttime</i> ConditionExpression is required in all queries.</p>
Body	<pre> <queryexpression> <filterexpression> <conditionexpression> <attribute> <attributetype>userid</attributetype> </attribute> <comparison>equal</comparison> <attributevalue> <value> 8958b4fb-cf9c-461e-a3c9-8c517845d128 </value> </attributevalue> </conditionexpression> <conditionexpression> <attribute> <attributetype>starttime</attributetype> </attribute> <comparison>between</comparison> <attributevalue> <value>2015-01-01 00:00:00</value> </attributevalue> <attributevalue> <value>2016-01-02 00:00:00</value> </attributevalue> </conditionexpression> <logicaloperator>and</logicaloperator> </filterexpression> <recordcount>0</recordcount> <orderby> <column> <attributecolumn> <attributetype>recid</attributetype> </attributecolumn> </column> <priority>0</priority> <order>asc</order> </orderby> </pre>

Example

```
</queryexpression>
```

<http://mygateway.local/query>

Return XML
Sample

```
<ArrayOfCallRecordCollection>
<CallRecordCollection count="2">
  <callrecord>
    <logicalcall>true</logicalcall>
    <callsegment>
      <segmentnumber>1</segmentnumber>
      <recid>2269831</recid>
    </callsegment>
    <attributes>
      <Key><type>recid</type></Key>
      <Value><string>2269831</string></Value>
      <Key><type>trunk</type></Key>
      <Value><string>94311</string></Value>
      <Key><type>callid</type></Key>
      <Value><string>478707</string></Value>
      ...
    </attributes>
  </callrecord>
  <callrecord>
    <logicalcall>false</logicalcall>
    <attributes>
      <Key><type>recid</type></Key>
      <Value><string>2269831</string></Value>
      <Key><type>trunk</type></Key>
      <Value><string>94311</string></Value>
      <Key><type>callid</type></Key>
      <Value><string>478707</string></Value>
      ...
    </attributes>
  </callrecord>
</CallRecordCollection>
</ArrayOfCallRecordCollection>
```

Return Definitions

ArrayOfCallRecordCollection: This is a collection of calls that match the conditions.

CallRecordCollection: This is collection of individual segments of a call that make up an entire call. A new segment is generated when a call get transferred between devices.

CallRecord: The details of an individual segment of a call.

LogicalCall: If a call is the logical call then this is a summarised version of the entire call.

Attributes: This contains key value pairs for details of the specific call record. The specific values available are dependent on the type of PBX system that is being used.

/endpoint/{number}/compliancemonitor/{instance}/add?timeout={timeout}

Description	Sets a compliance monitor for this endpoint instance. When a monitor is set the call recording will be paused until the compliance monitor is removed or expires. To keep the call paused the monitor must be resent before the expiry of the compliance keep alive interval. The expiry time is set by the timeout value or if not provided uses the server default. When this does elapse the call will be automatically resumed and recorded.
URL Structure	http://{gateway}/endpoint/{number}/compliancemonitor/{instance}/add?timeout={timeout}
Method	POST
Parameters	<i>number</i> : The endpoint device number. <i>instance</i> : This needs to be a unique identifier for this set of monitors. <i>timeout</i> : Optional value to set the expiry timeout for this set of monitors.
Body	If any of the monitors are already present then the relevant timeout value will be updated. <pre><ArrayOfString> <string>http://www.matching.moitor.one</string> <string>http://www.matching.moitor.two</string> </ArrayOfString></pre>
Example	http://mygateway.local/endpoint/1803/compliancemonitor/1001/add
Return XML Sample	
Exceptions	404 Not Found: If the endpoint number is invalid or does not exist.

/endpoint/{number}/compliancemonitor/{instance}/remove

Description	Removes a compliance monitor for this endpoint instance. This will then resume the recording of the active call.
URL Structure	http://{gateway}/endpoint/{number}/compliancemonitor/{instance}/remove
Method	POST
Parameters	<i>number</i> : The endpoint device number. <i>instance</i> : This needs to be a unique identifier for this set of monitors.

Body	<p>This is optional, if no monitors are specified then all the monitors for this endpoint/instance will be removed.</p> <pre><ArrayOfString> <string>http://www.matching.moitor.one</string> <string>http://www.matching.moitor.two</string> </ArrayOfString></pre>
Example	<code>http://mygateway.local/endpoint/1803/compliancemonitor/1001/remove</code>
Return XML Sample	
Exceptions	404 Not Found: If the endpoint number is invalid or does not exists.

/compliance/monitors

Description	Retrieves the list of compliance monitors that should be used to indicate a compliance monitor should be set.
URL Structure	<code>http://{gateway}/compliance/monitors</code>
Method	GET
Parameters	<i>none</i>
Returns	A string array of compliance monitors. Each item is a regular expression that can be used evaluate a matching compliance monitor.
Example	<code>http://mygateway.local/compliance/monitors</code>
Return XML Sample	<pre><ArrayOfString> <string>^http://www\.google\.co\.uk</string> <string>^(http:// https://) (.)* (xarios\.com)</string> </ArrayOfString></pre>
Return Definitions	This example returns a list of URI definitions that could be used to match the URLs that a user has open and use this to set a compliance monitor.

/compliance/configuration

Description	Retrieves the list of compliance configurations settings that have been configured on the server
URL Structure	<code>http://{gateway}/compliance/configuration</code>

Method	GET
Parameters	<i>none</i>
Returns	A configuration array of settings. Each item is a name value pair with a description that details the value and use of each setting.
Example	http://mygateway.local/compliance/configuration
Return XML Sample	<pre> <ArrayOfConfiguration> <Configuration> <Name>KeepAliveInterval</Name> <Value xsi:type="xsd:int">1000</Value> <Description>The time period in milliseconds that should be used to update the compliance monitors. </Description> </Configuration> <Configuration> <Name>SendMatchingMonitors</Name> <Value xsi:type="xsd:boolean">>true</Value> <Description>This sets if the client should send all the matching monitors in the body when calling /endpoint/{number}/compliancemonitor. </Description> </Configuration> </ArrayOfConfiguration> </pre>

/endpoint/compliance/references

Description	Retrieves the list of active compliance monitors for all endpoints. This provides details of the current monitors in real time.
URL Structure	http://{gateway}/endpoint/compliance/references
Method	GET
Parameters	<i>none</i>
Returns	A configuration array of references. Each references contains the details of the specific compliance monitor.
Example	http://mygateway.local/endpoint/compliance/references
Return XML Sample	<pre> <ArrayOfReference> <Reference> <Instance> <EndpointNumber>1803</EndpointNumber> <InstanceID> 495ae3e4-00ef-4d58-b037-a39ef75877dd </InstanceID> <IPAddress>172.19.20.88</IPAddress> <Port>65097</Port> <UserAgent> Mozilla/5.0 (Windows NT 6.1; rv:18.0) Gecko/20100101 Firefox/18.0 </UserAgent> </Instance> </Reference> </ArrayOfReference> </pre>

```
<Timestamp>2013-02-01T10:47:38.1073333+00:00</Timestamp>
<Timeout>15000</Timeout>
</Instance>
<MonitorReference>
  https://www.google.co.uk/
</MonitorReference>
<Timestamp>2013-02-01T10:47:38.1073333+00:00</Timestamp>
<Timeout>10000</Timeout>
</Reference>
</ArrayOfReference>
```

Return Definitions

EndpointNumber: The endpoint number.

InstanceID: The unique instance ID for this connection.

IPAddress: The IP address the client is connecting from.

Port: The TCP port the client is connecting from.

UserAgent: The User Agent string that determines the type of client.

Timestamp: The date and time when the client last connected.

Timeout: The amount of time in seconds as to when this monitor reference will expire.

MonitorReference: The monitor reference details for this instance, usually this is a URL that is currently open.

Utility Resources

/utility/newguid	
Description	Returns a new GUID string value.
URL Structure	<code>http://{gateway}/utility/newguid</code>
Method	GET
Parameters	<i>none</i>
Returns	A GUID string value.
Example	<code>http://mygateway.local/utility/newguid</code>
Return XML Sample	<code><string>E302732C-DA86-4676-A4D6-2E5A529A2B65</string></code>
Return Definitions	This example returns a new GUID string.

5. ActiveX Control

The features available within the ActiveX control are based upon the resources available in the Web service.

Requirements

The ActiveX control is a Microsoft Net 4.0 assembly that exposes a COM interface to enable the relevant methods, properties and events to be accessed. The ActiveX controls requires the .NET Framework 4 Client Profile to be installed alongside the ActiveX control.

The ActiveX connection details are shown below.

Connection Details	
ProgID	Xarios.Communication.Client.ActiveX
ClassID	0E7BC8C7-A03A-492A-BBE3-197106FFE454

Properties

There are several properties available that can be accessed via the ActiveX control to be able to configure the connection to the Communications Gateway.

string CommunicationGateway { get; set; }	
Details	The URI of the Communications Gateway, this needs to be set before any other methods are called.
Example	http://1.1.1.1:8088/

string ApplicationID { get; set; }	
Details	The Application ID for this connection to the Communications Gateway, this needs to be set before any other methods are called. The Application ID needs to have been pre-configured on the Communications Gateway to allow the connection to be authorised.
Example	D824ACEF-FF3F-41BE-82B3-80C92B0A7D7F

Methods

Each of the methods are mapped onto an associated Webs service call. See the relevant Web service documentation for details for valid parameters and return values.

string Endpoint_Get(string Number)

Web service	/endpoint/{number}
-------------	--------------------

void Endpoint_SetRecordingState (string Number, string State)

Web service	/endpoint/{number}/recordingstate?state={state}
-------------	---

void Endpoint_Tag(string Number, int Field, string Value)

Web service	/endpoint/{number}/tag?field={field}&value={value}&append={append}
-------------	--

void Endpoint_SetCustomTimeline(string Number, string Value)

Web service	/endpoint/{number}/tag?field={field}&value={value}
-------------	--

6. Microsoft .Net DLL

The features available within the Microsoft .Net DLL are based upon the resources available in the Web service.

Requirements

The Microsoft .Net DLL is a Microsoft Net 4.0 assembly that enables the relevant methods, properties and events to be accessed. The Microsoft .Net DLL assembly requires the .NET Framework 4 Client Profile to be installed alongside.

The Microsoft .Net DLL connection details are shown below.

Connection Details	
Namespace	Xarios.Communication.Client
Assembly	Xarios.Communication.Client.dll

Properties

There are several properties available that can be accessed via the assembly to be able to configure the connection to the Communications Gateway.

string CommunicationGateway { get; set; }	
Details	The URI of the Communications Gateway, this needs to be set before any other methods are called.
Example	http://1.1.1.1:8088/

string ApplicationID { get; set; }	
Details	The Application ID for this connection to the Communications Gateway, this needs to be set before any other methods are called. The Application ID needs to have been pre-configured on the Communications Gateway to allow the connection to be authorised.
Example	D824ACEF-FF3F-41BE-82B3-80C92B0A7D7F

Methods

Each of the methods are mapped onto an associated Webs service call. See the relevant Web service documentation for details for valid parameters and return values.

```
string Endpoint_Get(string Number)
```

Web service	/endpoint/{number}
-------------	--------------------

```
void Endpoint_SetRecordingState (string Number, string State)
```

Web service	/endpoint/{number}/recordingstate?state={state}
-------------	---

```
void Endpoint_Tag(string Number, int Field, string Value)
```

Web service	/endpoint/{number}/tag?field={field}&value={value}&append={append}
-------------	--

7. Result Codes

Code	Reason
1000	Generic error
1001	Unspecified error
1002	Endpoint not found
1003	Unable to change recording state
1004	Unknown request
1102	Invalid field number
1103	No call in progress
1104	Invalid recording state
1105	Invalid Parameter
1106	Unable to set timeline event
1107	User not found

8. URL Formats

The call recorder website can be accessed using URLs for direct playback for specific calls. For example using the *recid* or *logicalcallid* the user could click onto a URL that could take them to the playback page to listen to the recording. This uses the authentication built into the system to check that the user is allowed access to this call. If they are not logged in they will be presented with the standard login page and after they have entered their details they will be redirected to the relevant playback page.

The following formats are supported:

- Using the *recid*:
 - `http://{callrecorder}/default.aspx?recid={recid}`
- Using the *logicalcallid*:
 - `http://{callrecorder}/default.aspx?lclid={logicalcallid}`