INSTRUCTION MANUAL

VISION2 API REFERENCE GUIDE

VERSION 8.4

AV FOR AN IT WORLD®

IMPORTANT SAFETY INSTRUCTIONS

1. READ these instructions.
2. KEEP these instructions.
3. HEED all warnings.
4. FOLLOW all instructions.
5. DO NOT use this apparatus near water.
6. CLEAN ONLY with dry cloth.
7. DO NOT block any ventilation openings. Install in accordance with the manufacturer's instructions.
8. DO NOT install near any heat sources such as radiators, heat registers, stoves, or other apparatus (including amplifiers) that produce heat.
9. DO NOT defeat the safety purpose of the polarized or grounding type plug. A polarized plug has two blades with one wider than the other. A grounding type plug has two blades and a third grounding prong. The wider blade or the third prong are provided for your safety. If the provided plug does not fit into your outlet, consult an electrician for replacement of the obsolete outlet.
10. PROTECT the power cord from being walked on or pinched, particularly at plugs, convenience receptacles, and the point where they exit from the apparatus.
11. ONLY USE attachments/accessories specified by the manufacturer.
12. USE ONLY with a cart, stand, tripod, bracket, or table specified by the manufacturer, or sold with the apparatus. When a cart is used, use caution when moving the cart/apparatus combination to avoid injury from tip-over.
13. UNPLUG this apparatus during lightning storms or when unused for long periods of time.
14. REFER all servicing to qualified service personnel. Servicing is required when the apparatus has been damaged in any way, such as power-supply cord or plug is damaged, liquid has been spilled or objects have fallen into the apparatus, the apparatus has been exposed to rain or moisture, does not operate normally, or has been dropped.
15. DO NOT expose this apparatus to dripping or splashing and ensure that no objects filled with liquids, such as vases, are placed on the apparatus.
16. To completely disconnect this apparatus from the AC Mains, disconnect the power supply cord plug from the AC receptacle.
17. Where the mains plug or an appliance coupler is used as the disconnect device, the disconnect device shall remain readily operable.
18. DO NOT overload wall outlets or extension cords beyond their rated capacity as this can cause electric shock or fire.

The exclamation point, within an equilateral triangle, is intended to alert the user to the presence of important operating and maintenance (servicing) instructions in the literature accompanying the product.

The lightning flash with arrowhead symbol within an equilateral triangle is intended to alert the user to the presence of uninsulated "dangerous voltage" within the product's enclosure that may be of sufficient magnitude to constitute a risk of electrical shock to persons.

ESD Warning: The icon to the left indicates text regarding potential danger associated with the discharge of static electricity from an outside source (such as human hands) into an integrated circuit, often resulting in damage to the circuit.

WARNING: To reduce the risk of fire or electrical shock, do not expose this apparatus to rain or moisture.

WARNING: No naked flame sources - such as candles - should be placed on the product.

WARNING: Equipment shall be connected to a MAINS socket outlet with a protective earthing connection.

WARNING: To reduce the risk of electric shock, grounding of the center pin of this plug must be maintained.

COPYRIGHT NOTICE

LIABILITY NOTICE

No patent liability is assumed with respect to the use of information contained herein. While every precaution has been taken in the preparation of this publication, AMX assumes no responsibility for error or omissions. No liability is assumed for damages resulting from the use of the information contained herein. Further, this publication and features described herein are subject to change without notice.

AMX WARRANTY AND RETURN POLICY

The AMX Warranty and Return Policy and related documents can be viewed/downloaded at www.amx.com.

Table of Contents

Using the API

Overview

The control API for Vision^2 enables external control of the system. The control API is accessible from the Master server in the Vision^2 system, and the Vision^2 system manages sending subsequent requests to other servers in the system. A programmer using the API should not have to know which server to send an API request to, as all requests go to the Master server. This document pertains to Vision^2 version 8.4.

Controlling the Amino Set-Top Box

The following sections detail how to use the Vision^2 control API with the Amino set-top box.

Forcing a Set-Top Box to Play a Specific Live Channel

Before using the API, a list of available set-top boxes must first be obtained using the *v2Master_STB_Amino_GetSTBs* command.

This command is used by accessing a Vision^2 server from a web browser. For example, the command can be used in the following URL:

```
http://<server>/v2services/v2_API/v2_APIAdminService.asmx/
v2Master_STB_Amino_GetSTBByFirstLetterInName?sStartLetter=*
```

where <server> is the name of the Master server.

This command returns XML code similar to the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
    <stbs localip="43.192.168.100">
      <stb name="Main Reception" ip="43.192.168.101" />
      <stb name="Board Room" ip="43.192.168.102" />
       <stb name="Canteen" ip="43.192.168.111" />
<stb name="Common Area 1" ip="43.192.168.112" />
<stb name="Common Area 2" ip="43.192.168.113" />
  </stbs>
</v2>
```

From this XML code, extract the name and IP attributes to provide a means of selecting the desired STB:

"Main Reception" = "43.192.168.101"

"Board Room" = "43.192.168.102"

Next, a list of live channels which the Amino can play must be obtained by using the *v2Master_GetLiveChannelListLastUpdateTime* command. For example:

```
http://server/v2services/v2_API/v2_APIUserService.asmx/v2Master_GetLiveChannelListLastUpdateTime
?sWantWMV=false&sWantMPEG2=true&sWantH264=true
```

Note: *The livegroup= must be included at the end of this url but can be left empty. If not you will get a http 500 error.*

This command returns XML code similar to the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
  <livechannels>
    <channel
     type="v2DVB"
     id="9018:16384:17024"
     name="BBC Parliament"
     address="234.1.0.2"
     port="5500"
     url="videocastmcast://234.1.0.2:5500/BBC Parliament"
     level="0"
     unmanaged="no"
     />
    <channel
     type="v2DVB"
     id="9018:16384:17023"
     name="BBC One"
     address="234.1.0.3"
     port="5500"
     url="videocastmcast://234.1.0.3:5500/BBC One"
     level="0"
     unmanaged="no"
    />
  </livechannels>
  <secure>true</secure>
</v2>
```

From this code, extract the name attribute to provide a means of selecting the desired channel:

"BBC Parliament"

"BBC One"

For illustration purposes, assume the user selects "Main Reception" set-top box and the "BBC One" channel. The user must then to make the change call to the *v2Master_STB_Amino_SetLiveChannel* command. For example:

```
http://<server>/v2services/v2_API/v2_APIAdminService.asmx/
v2Master_STB_Amino_SetLiveChannel?sSTBIPAddress=43.192.168.110&sChannelIndex=1
```

Forcing a Specific Set-Top Box to Play a Specific Video

Obtain a list of the available set-top boxes as described in the *Forcing a Set-Top Box to Play a Specific Live Channel* section on page 1. Next, provide functionality to enable the user to choose the video to play. See the *Vision2 RenderThumbs Explorer* section on page 21 for information on how to create a user interface to play a video.

When the user has selected the video to play, use the *v2Master_STB_Amino_PlayVideo* command to start playback. For example:

```
http://<server name>/v2services/v2_API/v2_APIAdminService.asmx/
v2Master_STB_Amino_PlayVideo?sSTBIPAddress=192.168.0.14&sArchiveID=201101020304&sVideoID=150002
```

Enabling and Disabling Managed Live Channels

To control which channels are broadcasting on a Vision2 system the first thing needed is a list of all the available managed live channels, which can be found using the v2Master_GetFullChannelList command. For example:

```
http://server/v2services/v2_API/v2_APIAdminService.asmx/v2Master_GetFullChannelList
```

This command returns XML code similar to the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
  <channels>
    <channel
       type="v2DVB"
       name="BBC THREE"
       id="9018:4100:4351"
       transmitting="false" />
    <channel
       type="v2DVB"
       name="BBC NEWS"
       id="9018:4100:4415"
       transmitting="true" />
    <channel
       type="v2Live"
       name="MPEG Encoder"
       id="20110014111920046"
       transmitting="false" />
    <channel
       type="v2Producer"
       name="Producer MPEG 1"
       id="20110014111920687"
       transmitting="false" />
    <channel
       type="v2ReflectorWMV"
       name="WMV Reflector 1"
       id="20110014111921218"
       transmitting="true" />
    <channel
       type="v2ProducerWMV"
       name="Producer WMV 1"
       id="20110014111921593"
       transmitting="false" />
  </channels>
</v2>
```

Next, using the Live, Producer and Reflector Start/Stop Commands, convert this code into a list from which the user can select a channel and change the transmitting status as necessary, depending on the channel type needed to make the correct call.

- *v2DVB_StartBroadcast* on page 19
- *v2Live_StartMPEGEncoder* on page 13
- *v2Producer_StartBroadcast* on page 15
- *v2Reflector_StartBroadcast* on page 18
- *v2DVB_StopBroadcast* on page 19
- *v2Live_StopMPEGEncoder* on page 13
- *v2Producer_StopBroadcast* on page 15
- *v2Reflector_StopBroadcast* on page 18

Generate Session Id

Beginning with version 8.2, many services have user names and passwords required for authentication. Since no username or password is passed between Netlinx devices, the SessionID is used by the locking mechanism to determine if the service is being used or not. For example if User A is starting a broadcast, the producer service will be locked preventing any other user from getting access to that service. This is achieved by using the SessionID

SessionID is a GUID (*in the format xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx*) which can be generated using either one of the following websites

http://www.guidgen.com/
http://createguid.com/

Or can be generated using the downloadable file using http://guid.codeplex.com/releases/

Once a session Id is obtained together with a NETLINX generated username, this information is used to lock whichever NETLINX API call is being used. This prevents other users from gaining access to a service that has been locked. For example, if a user is using the record service and has started a recording, then a lock is obtained. If the same user then stops the recording the lock is removed. No other user can access the record service whilst a record is taking place.

The following API calls work in tandem when acquiring and releasing locks:

- "V2Live_StartMpegEncoder and V2Live_stopMpegEncoder
- "V2Producer_StartBroadcasting and V2Producer_StopBroadcasting
- "V2Record_StartRecording and V2Record_StopRecording/V2Record_CancelRecording (For Manual Recording)
- "V2Record_RestartRecording and V2Record_StopRecording/V2Record_CancelRecording (For Manual recording)
- "V2Record_EnableRecorder (For all types of recording) - by passing in a true will lock the service and by passing in false will unlock.
- "V2Reflector_StartBroadcast and V2Reflector_StopBroadcast

When recording content the encoder must be enabled. Once enabled, if the recording is a manual recording, then you can start and then cancel the recording or start and then stop the recording.

Netlinx Generated User Name

In addition to the generated session ID, a generated user name is also required and is used for identification purposes. So for example, for a Netlinx device with the username as Netlinx001X the locking mechanism will indicate that user Netlinx001X has a lock on the service to another user who tries to access it.

Enabling IIS Security for the Administrator API

With IIS security enabled, a block of code attempting to access an API function will require an Administrator log in. Perform the following steps to enable IIS security for the Administrative API:

1. On the Master Server, click the following path: Start Menu > Administrative Tools > Internet Information Services (IIS) Manager. The Internet Information Services (IIS) Manager opens.
2. In the tree view on the left pane, navigate to *server name* > Web Sites > Default Web Site > v2Services > V2_API.
3. In the right pane, right-click v2_APIAdminServicebrowse.asmx and select **Properties**.
4. Select the **File Security** tab.
5. In the Authentication and access control section, click **Edit**.
6. In the Authentication Methods dialog box, uncheck **Enable anonymous access**, and click **OK**.
7. Click **Apply**, then click **OK** to close the window.

Note: *This change only needs to be made on the Master server.*

API Commands

Main Control API

The main control API consists of a standard XML based web service. An interactive means of testing the API is available at:

```
http://_server_name_/v2Services/v2_API/v2_APIAdminServicebrowse.asmx
```

Accessing Video On Demand Content

The Vision^2 RenderThumbs Explorer is available as a way of simplifying user access to content from a Vision^2 archive. The Vision^2 RenderThumbs Explorer provides the following advantages:

- Much simpler than creating a custom user interface
- Single call provides either data or a rendered user interface as an image
- Data is available as either XML or JSON
- Rendered user interface is template to provide both customization and uniformity
- Even if a custom user interface is not needed, the returned XML | JSON is much easier to process than accessing the raw content.

This API is used to provide the user interface for the new Amino 140H STB and for the new Tablet PC support for Vision^2. See the *Vision2 RenderThumbs Explorer* section on page 21 for information.

Note: *In the API list, calls marked with * are expected to be useful in a touch panel interface.*

Note: *STB devices that send an "offline" status to Vision2 will appear as offline in the STB table and the icon will turn grey.*

V2_APIAdminService

API calls should always include all available variables unless noted (Optional), and always be in the order specified by this document.

Note: *For all services that use session ID and Username, please refer to Generate Session Id* on page 3 *and Netlinx Generated User Name* on page 3*.*

V2Master_KeepAlive

Runs as background process and keeps the session alive after each interval. It is recommended to be called after every 5 minutes.

Parameters	
SessionId	User session id in guid format
UserName	Netlinx generated username

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
<result>OK</result>
</v2>
```

Master Commands

The following section lists the API commands available for the Master.

v2Master_STB_Amino_GetSTBs

Get an XML list of all of the Set-Top Boxes.

Sample URL

```
http://server/v2services/v2_API/v2_APIAdminService.asmx/v2Master_STB_Amino_GetSTBs
```

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
    <stbs localip="43.192.168.100">
      <stb
  name="Main Reception"
  status="playing video on demand"
  playing="Welcome to AMX"
playingraw="rtsp://43.192.168.100:8554/H:\vcContent\Archive1\1500\68\ts\high\Video.ts"
  ip="43.192.168.110"
  pending="nothing"
  pendingparams="-1"
  screensaver="0"
  arc="true" />
      <stb
  name="Board Room"
  status="idle"
  playing=" "
  playingraw=" "
  ip="43.192.168.111"
  pending="nothing"
  pendingparams="-1"
  screensaver="0"
  arc="true" />
  </stbs>
</v2>
```

Program Attributes	
name	The name given to the STB in the management
status	What the STB is currently doing. Can be: • "playing live channel" • "playing video on demand" • "idle".
playing	If playing then the name of the video or live channel
playingraw	The video URL currently being accessed
ip	The IP Address of the STB
pending	The remote command currently being processed. Can be: • ChangeChannel • PlayVOD • ARC.
pendingparams	Parameter values for any pending command
Screensaver	Not used
arc	If true means the STB can Access Restricted Content

v2Master_STB_Amino_GetSTBsByFirstLetterInName

Get an XML list of all of the Set Top Boxes whose name begins with the letter specified. Note this call only returns the STBs name and IP Address.

Sample URL

```
http://server/v2services/v2_API/v2_APIAdminService.asmx/
v2Master_STB_Amino_GetSTBsByFirstLetterInName?sStartLetter=c
```

Parameters	
sStartLetter	The start letter if the name. Case is ignored. If this is '*' then all STBs are returned.

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
    <stbs localip="43.192.168.100">
      <stb name="Canteen" ip="43.192.168.111" />
<stb name="Common Area 1" ip="43.192.168.112" />
<stb name="Common Area 2" ip="43.192.168.113" />
  </stbs>
</v2>
```

Program Attributes	
name	The name given to the STB in the management UI
ip	The IP Address of the STB

v2Master_STB_Amino_GetStatus

Obtains the current status of the STB specified by its IP Address

Sample URL

```
http://server/v2services/v2_API/v2_APIAdminService.asmx/
v2Master_STB_Amino_GetStatus?sSTBIPAddress=43.192.168.110
```

Parameters	
sSTBIPAddress	The IP Address of the STB to obtain the status from

Sample XML:

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
  <stb
       name="Main Reception"
       status="playing video on demand"
       playing="Toxic"
       playingraw="rtsp://43.192.168.100:8554/
       H:\vcContent\Archive1\1500\68\ts\high\Video.ts"ip="43.192.168.110"
       ip="43.192.168.110"
       pending="nothing"
       pendingparams="-1"
       screensaver="0"
       arc="true" />
</v2>
```

Note: *See the v2Master_STB_Amino_GetSTBs* section on page 5 *for an explanation of the returned XML.*

v2Master_STB_Amino_SetLiveChannel

Forces the STB to display the selected live channel.

Note: *The list of suitable live channels can be obtained by calling:*

```
http://<servername>/v2Servers/v2_API/v2_APIAdminService.asmx/v2Master_STB_Amino_GetLiveChannelList
     ?sWantWMV=false
     &sWantMPEG2=true
     &sWantH264=true
```

Sample URL

```
http://server/v2services/v2_API/v2_APIAdminService.asmx/v2Master_STB_Amino_SetLiveChannel
   ?sSTBIPAddress=43.192.168.110
   &sChannelIndex=2
   &LiveGroup=0
   &WaitForResponse=True
```

Parameters	
sSTBIPAddress	The IP Address of the STB to obtain the status from
sChannelIndex	The index of the channel to play. By calling index channel GetLiveChannel the channel index can be obtained from here or from the list returned. The first channel in the list starts with the ChannelIndex=0
LiveGroup	Which LiveGroup to base channel selection on. (0 if not using feature)
WaitForResponse	(Optional, default "True") Set as "False" to skip Amino confirmation.

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
</v2>
```

v2Master_STB_Amino_SetLiveChannelByID

Forces the STB to display the selected live channel

Sample URL

```
http://server/v2services/v2_API/v2_APIAdminService.asmx/v2Master_STB_Amino_SetLiveChannelByID
        ?sSTBIPAddress=43.192.168.110
        &sChannelID=20110701095642500
        &sBookmark=false
        &LiveGroup=0
        &WaitForResponse=True
```

Parameters	
sSTBIPAddress	The IP Address of the STB to obtain the status from. If the IP Address is 'ALLSTBS' then all the STBs on the system will be updated.
sChannelID	The ID of the channel to play
sBookmark	If 'true' then the current channel or VOD status is saved for later use by a call to v2Master_STB_Amino_GoBack
LiveGroup	Which LiveGroup to base channel selection on. (0 if not using feature)
WaitForResponse	(Optional, default "True") Set as "False" to skip Amino confirmation.

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
</v2>
```

Note: *The list of suitable live channels can be obtained by calling:*

```
http://<servername>/v2Servers/v2_API/v2_APIAdminServicebrowse.asmx v2Master_GetLiveChannelList
     ?sWantWMV=false
     &sWantMPEG2=true
     &sWantH264=true
```

V2Master_STB_Amino_SetARC

Allows the Amino access to restricted content.

Parameters	
sSTBIPAddress	The IP Address of the STB to set. "ALLSTBS" is a valid input.
sARCValue	Can either be "true" or "false"

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
</v2>
```

v2Master_STB_Amino_PlayVideo

Forces the STB to play a Video On Demand Title

Sample URL

```
http://<server name>/v2services/v2_API/v2_APIAdminService.asmx/
v2Master_STB_Amino_PlayVideo?sSTBIPAddress=192.168.0.14
   &sArchiveID=cc4597b2-e336-424f-bcca-a728c1e0c5d9
   &sVideoID=a12a4316-e036-436e-a1fb-0e388d06d784
   &sBookmark=false
   &WaitForResponse=True
```

Parameters	
sSTBIPAddress	The IP address of the STB to obtain the status from
sArchiveID	The ID of the Archive Service on which the video resides
sVideoID	The ID of the video to play
sBookmark	Whether to save the previous state for GoBack functionality
WaitForResponse	(Optional, default "True") Set as "False" to skip Amino confirmation.

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
</v2>
```

v2Master_STB_Amino_SendRS232

Send a string via RS-232 on the Amino140H (only) and wait for the reply.

Sample URL:

```
http://<server>/v2services/v2_API/v2_APIAdminService.asmx/v2Master_STB_Amino_SendRS232
   ?sSTBIPAddress=43.192.168.110
   &sBaudRate =19200
   &sWantBinary=false
   &sData= ka 01 01\n
   &WaitForResponse=True
```

Parameters	
sSTBIPAddress	The IP address of the set-top box to obtain the status from
sBaudRate	Must be 9600 or 19200
sWantBinary	Must be false in this release
sData	The string to send (note that \r and \n will be converted to actual ASCII and any escaped characters (e.g. %03) will also be converted.)
WaitForResponse	(Optional, default "True") Set as "False" to skip Amino confirmation.

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>a 01 OK01\n</result>
</v2>
```

As of Version 8.2.4329.1835, multiple commands can be sent in the sData parameter by following this pattern:

```
sData=Command~DelayInSeconds~Command~DelayInSeconds~Command
```

Command refers to "ka 01 01\n" in the sample and DelayInSeconds is the number of seconds to wait before executing the next command in sequence. The sequence MUST end in a command and will only continue executing on success response from previous command. Results will include responses for each command up to the last command sent/or last successful command, separated by ~'s.

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>a 01 OK01\n~a 01 OK01\n~a 01 OK01\n</result>
</v2>
```

Note: *\r and \n in sData will be converted to ASCII and that any escaped characters (e.g %03) will also be converted.*

v2Master_STB_Amino_GoBack

Forces the STB to return to what they were watching when a command was called with *sBookmark=true*.

Sample URL

```
http://server/v2services/v2_API/v2_APIAdminServicebrowse.asmx/v2Master_STB_Amino_GoBack
   ?sSTBIPAddress=43.192.168.110
   WaitForResponse=True
```

Parameters	
sSTBIPAddress	The IP Address of the STB to obtain the status from. If the IP Address is ‘ALLSTBS’ then all the STBs on the system will be updated.
WaitForResponse	(Optional, default "True") Set as "False" to skip Amino confirmation.

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
</v2>
```

v2Master_GetAPIVersion

Returns the current version of the API.

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
  <version>
     <major>8.4</major>
     <autobuild>17955941</autobuild>
  </version>
  <secure>true</secure>
</v2>
```

Note: *Secure is set true if Vision^2 User Access Control is enabled on the system and offers an easy method to check if user log in is required to access content.*

v2Master_GetFullChannelList

Obtain a list of all of the Vision² channels even the ones not transmitting.

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
 <result>OK</result>
 <channels>
   <channel type="v2DVB"
     name="BBC THREE"
     id="9018:4100:4351"
     transmitting="false" />
   <channel type="v2DVB"
     name="BBC NEWS"
     id="9018:4100:4415"
     transmitting="true" />
   <channel type="v2DVB"
     name="BBC TWO"
     id="9018:4100:4228"
     transmitting="false" />
   <channel type="v2DVB"
     name="BBC ONE"
     id="9018:4100:4164"
     transmitting="true" />
   <channel type="v2DVB"
     name="301"
     id="9018:4100:7168"
     transmitting="false" />
   <channel type="v2Live"
     name="MPEG Encoder"
     id="20110014111920046"
     transmitting="false" />
   <channel type="v2Producer"
     name="Producer MPEG 1"
     id="20110014111920687"
     transmitting="false" />
   <channel type="v2ReflectorWMV"
     name="WMV Reflector 1"
     id="20110014111921218"
     transmitting="true" />
   <channel type="v2ProducerWMV"
     name="Producer WMV 1"
     id="20110014111921593"
     transmitting="false" />
  </channels>
</v2>
```

v2Master_GetDiagnostics

Retrieves the full diagnostics from the server.

Sample XML

```
<?xml version="1.0" encoding="UTF-8"?>
<v2>
    <result>OK</result>
    <os>
        <name>Microsoft Windows 7 Professional </name>
        <version>Service Pack 1</version>
        <serial>00000-000-0000000-00000</serial>
    </os>
    <memory>
        <total>16720144</total>
        <free>10231420</free>
    </memory>
    <CPUs>
        <CPU>
            <loadpercentage>0</loadpercentage>
        </CPU>
    </CPUs>
    <nics>
        <name> Intel[R] 82579LM Gigabit Network Connection </name>
        <connectionspeed>100000000</connectionspeed>
        <megabitsreceiving>366008810</megabitsreceiving>
        <megabitssending>17771922</megabitssending>
    </nics>
</v2>
```

v2Master_GetListOfServers

Obtain a list of all of the Vision2 servers in the system and determine if they are online.

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
   <servers>
      <server
         name="WIN2K3SERVER"
         webserviceurl="http://WIN2K3SERVER/v2Services/"
         online="true" />
      <server
         name="LENOVO"
         webserviceurl="http://LENOVO/v2Services/"
         online="false" />
  </servers>
</v2>
```

Note: *If many of the Vision^2 Servers are offline then this call may take some time before it returns as the system times-out trying to contact an offline server.*

v2Master_GetLiveChannelList

Obtains a full list of available Active Live channels both Managed and Unmanaged.

Parameters	
sWantWMV	'true' if you want the list to include Windows Media Format Channels
sWantMPEG2	'true' if you want the list to include MPEG-2 (Video) Format Channels
sWantH264	'true' if you want the list to include h.264 (Video) Format Channels
type	Channel

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
  <v2>
    <result>OK</result>
    <channels>
       <channel
         type="v2DVB"
         id="9018:16384:17024"
         name="BBC Parliament"
         address="234.1.0.2"
         port="5500"
         url="videocastmcast://234.1.0.2:5500/BBC Parliament"
         level="0"
         unmanaged="false"
    </channel>
   </channels>
</v2>
```

v2Master_GetListOfServices

Obtains a list of selected service types on the entire Vision2 system.

Parameters	
sServiceType	The type of services to return. This can be one of the following values; • v2Archive • v2Record • v2DVB • v2Producer • v2Live • ALL - Returns every service on the system

Sample XML

```
 <?xml version="1.0" encoding="utf-8" ?>
  <v2>
    <result>OK</result>
  <services>
  <service
         servicetype="v2DVB"
         id="20110014111919578"
         name="DVB 1"
         tuner="AMX DTV-TX01 DVB-T Multicast on NIC 2"
         captureepg="false"
         activated="true"
         licensed="true"
         capturenan="false"
         demomode="false"
         basemulticastaddress="234.1.0.0"
         port="5500"
         ttl="2"
         nic="1"
         frequency="Multicast 224.10.0.9"
         archivetype="LME"
         webserviceurl="http://WIN2K3SERVER/v2Services/"
         subtitles="false"
         secondaryaudio="false"
         buffersize="1316"
         scanfile="SINGLE_MUX_MULTICAST_DVB:224.10.0.9:1234" />
  </services>
</v2>
```

Note: *For most calls to the system, the service ID is required.*

v2Master_GetListOfServices_NameAndIDOnly

Obtains a list of selected service types on the entire Vision2 system returning only the 'name' and 'id' attributes.

Parameters	
sServiceType	The type of services to return. This can be one of the following values; • v2Archive • v2Record • v2DVB • v2Producer • v2Live • ALL - Returns every service on the system

Sample XML

```
 <?xml version="1.0" encoding="utf-8" ?>
   <v2>
         <result>OK</result>
   <services>
  <service id="cc4597b2-e336-424f-bcca-a728c1e0c5d9"
   name="Archive Service 1" />
         <service id="20110014111919578" name="DVB 1" />
   </services>
  </v2>
```

v2Master_GetLiveChannelListLastUpdateTime

Gets the time and date of the last time the live channel list was updated.

```
<?xml version="1.0" encoding="utf-8" ?>
 <v2>
         <result>OK</result>
         <updated>11/2/2013 12:13:53 AM</updated>
 </v2>
```

Live Commands

The following section lists the API commands available for the Live MPEG.

v2Live_StartMPEGEncoder

Starts the specified encoder transmitting.

Channel Attributes	
sServiceID	The service ID of the Live Service to start
SessionID	User session id in guid format
UserName	Netlinx generated username

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
</v2>
```

v2Live_StopMPEGEncoder

Stops the specified encoder transmitting.

Channel Attributes	
sServiceID	The service ID of the Live Service to stop
SessionID	User session id in guid format
UserName	Netlinx generated username

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
</v2>
```

Producer Commands

The following section lists the API commands available for the Producer service. All commands are valid MPEG producer channels.

v2Producer_GetSchedule

Obtains the XML schedule for the specified Producer channel.

Channel Attributes	
sServiceID	The service ID of the Producer Service.

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
  <week>3</week>
  <day>3</day>
  <schedule>
   <week id="0">
    <day id="0">
       <program name="Test Clips - Program News"
        duration="00:06"
        colourindex="1"
        id="207ea4c5-dfa1-4f77-b1b9-def8ed006bfb"
        uniqueid="entry0"
        starttimemins="383"
        endtimemins="389"
        start="06:22"
        end="06:28"
        url="H:\vcContent\Archive1\0000\05\TS\High\video.ts"
        server="V2-AMX-R720"
        archiveid="cc4597b2-e336-424f-bcca-a728c1e0c5d9"
        parentid="63380661-b79b-871e-cfd2-841a5705a98a" />
       <program name="Test Clips - Program Music"
        duration="00:06"
        colourindex="1"
        id="5cb77af2-1014-408a-8bba-7ceef213d788"
        uniqueid="entry1"
        starttimemins="389"
        endtimemins="395"
        start="06:28"
        end="06:34"
        url="H:\vcContent\Archive1\0000\03\TS\High\video.ts"
        server="V2-AMX-R720"
        archiveid="cc4597b2-e336-424f-bcca-a728c1e0c5d9"
        parentid="63380661-b79b-871e-cfd2-841a5705a98a" />
     </day>
    </week>
  </schedule>
 </v2>
```

Note: *'week' is current week in schedule and 'day' is the current day in the schedule.The schedule starts at Week 0 and ends at Week 3, days start at 0 (Sunday). If no content is scheduled for a week then there will be no week node. Also, if no content is scheduled then there will be no day node.*

Program Attributes	
name	Name of the video
duration	How long the video lasts
colourindex	Not used
id	The ID of the video within the Archive
uniqueid	A unique ID of this program within this schedule
starttimemins	Scheduled Start Time
endtimemins	Scheduled End time
start	Scheduled Start Time
end	Scheduled End time
url	The URL of the video

v2Producer_SetParameters

Sets the parameters for the specified Producer channel.

Parameters	
sServiceID	The service ID of the Producer Service
sIntermissionEnable	'true' to enable intermission
sIntermission	The URL (for Live channels) or path to the video file to be used as the source for the intermission.
sNICIndex	The index of the NIC to use for multicast (NB not the IP Address)
sMulticastAddress	The multicast address to use for the broadcast
sMulticastPort	The multicast port to use for the broadcast
sMulticastTTL	The multicast TTL to use for the broadcast
sBroadcastIsMulticast	Always 'true' for MPEG.
SessionID	User session id in guid format
UserName	Netlinx generated username

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
</v2>
```

v2Producer_StartBroadcast

Starts the selected producer service broadcasting and adds the channel to the playlists.

Parameters	
sServiceID	The service ID of the Producer Service
SessionID	User session id in guid format
UserName	Netlinx generated username

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
</v2>
```

v2Producer_StopBroadcast

Stops the selected producer service broadcasting and adds the channel to the playlists.

Parameters	
sServiceID	The service ID of the Producer Service
SessionID	User session id in guid format
UserName	Netlinx generated username

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
</v2>
```

Record Commands

The following section lists the API commands available for the Record service. These commands only controls a Record service that is configured and enabled as a Manual record service.

v2Record_StartRecording

Starts the selected record service recording.

Parameters	
sServiceID	The service ID of the Recording Service to start
sRecordingName	The name that appears in the Archive.
SessionID	User session id in guid format
UserName	Netlinx generated username

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
</v2>
```

v2Record_StopRecording

Stops the selected record service recording.

Parameters	
sServiceID	The service ID of the Recording Service to stop
SessionID	User session id in guid format
UserName	Netlinx generated username

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
</v2>
```

v2Record_CancelRecording

Stops the current recording, deletes the file from the disk and prevents it being archived. This call is only valid for v2Record services and not v2RecordWMV services.

Parameters	
sServiceID	The service ID of the Recording Service to cancel the recording of
SessionID	User session id in guid format
UserName	Netlinx generated username

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
</v2>
```

Note: *This call is only valid for v2Record services and not v2RecordWMV services.*

v2Record_EnableRecorder

Before setting the parameters, the recorder must be disabled via Vision2 or this command. This command will enable/disable the recorder. To disable, send the command with sEnabled = false. To enable the recorder again, send the command with sEnabled = true..

Parameters	
sServiceID	The service ID of the Recorder Service to enable/disable
sServiceName	The name of the Recorder Service to enable/disable
sEnabled	To enable the Recorder Service, sEnabled = “true” To disable the Recorder Service, sEnabled = “false”;
SessionId	User session id in guid format
UserName	Netlinx generated username

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
</v2>
```

v2Record_RestartRecording

Restarts the current manual recording and the current recording is overwritten.

Parameters	
sServiceID	The service ID of the Recording Service to start the recording of
SessionID	User session id in guid format
UserName	Netlinx generated username

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
</v2>
```

v2Record_SetParameters

Sets the parameters for the selected record service. Before setting the parameters, the recorder must be disabled via Vision2 or the *v2Record_EnableRecorder* command. Once the parameters are set, return to the *v2Record_EnableRecorder* command to enable the recorder..

For Manual type recordings, the *Segment Length* and *Keep Archive Days* parameters are not required and will be ignored. However, for Continuous type recordings, these two parameters are required and if missing, you will get an error message.

Parameters	
sSessionId	User session id in guid format
sServiceId	The service ID of the Recording Service
sUserName	Netlinx generated username
sNICindex	The index of the NIC to use to receive the stream
sRecordingType	Defines how the service operates • Manual = Recording is manually started and stopped by calls to the web service • Continuous = Recording is non-stop • Scheduled = Recordings are made to a schedule
sSegmentLength	If Record Type = 'Continuous', then a new file will automatically be created and added to the archive every 'Segment Length' minutes. If missing you will receive an error message. Ignore this parameter for Manual recordings.
sChannelName	The name of the channel being recorded
sKeepArchiveDays	If Record Type = 'Continuous', then how long to keep the recordings before they are automatically deleted from the Archive. If missing you will receive an error message. Ignore this parameter for Manual recordings.
Optional Parameters (for backwards compatibility)	
sArchiveServiceURL	The URL to the Archive Service
sArchiveToWriteTo	ID of the archive to be written to must be either a local archive or NAS Server archive. You cannot record on an archive on a different server from the record service.

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
</v2>
```

Reflector Commands

The following section lists the API commands available for the Reflector service.

v2Reflector_StartBroadcast

Starts the selected reflector service broadcasting.

Parameters	
sServiceID	The service ID of the Reflector service to start.
SessionID	User session id in guid format
UserName	Netlinx generated username

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
</v2>
```

v2Reflector_StopBroadcast

Stops the selected reflector service broadcasting.

Sample XML

Parameters	
sServiceID	The service ID of the Reflector service to stop
SessionID	User session id in guid format
UserName	Netlinx generated username

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
</v2>
```

v2Reflector_SetParameters

Sets the parameters for the specified Reflector service.

Parameters	
sServiceID	The service ID of the Reflector service to target
sSessionID	User session id in guid format
sUserName	Netlinx generated username
sIsMulticast	Set to 'true' for multicast or 'false' for unicast settings
sMulticastIp	The multicast IP to use for the reflector
sMulticastPort	The multicast port to use for the reflector
sMulticastTtl	The multicast TTL to use for the reflector
sNicIndex	The index of the NIC to use for the reflector (NB not the IP Address)
sUnicastUrl	The unicast Url to use for the reflector (e.g. http://192.168.1.1:8000/Reflector)
sUnicastPort	The unicast Port to use for the reflector
sMulticastAddress	The multicast address to use for the reflector
sSourceUrl	The source url to feed the reflector
sSourceType	Set to 'localv2' for Local Vision2 Channel or 'external' for an External Source
sNewServiceName	A new name to assign to this reflector service.

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
</v2>
```

v2Reflector_GetSettings

Returns the current Reflector settings

Parameters	
sServiceID	The service ID of the reflector service to target

Sample XML

```
<?xml version="1.0" encoding="UTF-8"?>
<v2>
     <result>OK</result>
     <params>
          <param type="true-false" selected="false" name="Enabled"/>
          <param type="text" selected="234.0.0.0:5500" name="Source URL"/>
          <param type="list" selected="" name="Source ID" selectby="value">
               <option name="Manual Unicast Source" id="internetunicastsource"
                value="internetunicastsource"/>
          </param>
          <param type="list" selected="0" name="NIC" selectby="index">
               <option name="192.168.1.1" value="192.168.1.1"/>
          </param>
          <param type="text" selected="234.0.0.0" name="Multicast Address"/>
          <param type="text" selected="5500" name="Multicast Port"/>
          <param type="text" selected="3" name="Multicast TTL"/>
          <param type="true-false" selected="true" name="Broadcast Is Mulicast"/>
          <param type="text" selected="" name="Unicast URL"/>
     </params>
</v2>
```

DVB Commands

The following section lists the API commands available for the DVB service.

v2DVB_StartBroadcast

Starts the selected DVB channel for broadcasting.

Parameters	
sServiceID	The channel ID to start
SessionID	User session id in guid format
UserName	Netlinx generated username

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
</v2>
```

Note: *Here the Channel ID is needed and not the Service ID as a DVB service can host many channels.*

v2DVB_StopBroadcast

Stops the selected DVB channel from broadcasting.

Parameters	
sServiceID	The channel ID to stop
SessionID	User session id in guid format
UserName	Netlinx generated username

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
</v2>
```

Note: *Here the Channel ID is needed and not the Service ID as a DVB service can host many channels.*

v2Archive Commands

The following section lists the API commands available for the v2Archive service.

v2Archive_GetDiskSpace

Retrieves the storage available and storage used of a given Archive

Parameters	
sServiceId	The service ID of the Archive Service

Sample XML

```
<?xml version="1.0" encoding="UTF-8"?>
<v2>
     <result>OK</result>
     <freespace>
          <entry data="160.0000" label="Free Space 160.00 GBytes"/>
          <entry data="340.0000" label="Used Space 340.00 GBytes"/>
     </freespace>
</v2>
```

v2Archive_DeleteVideo

Permanently deletes a video.

Parameters	
sArchiveId	The ID of the Archive Service on which the video resides
sCategoryId	The ID of the category containing the video
sVideoId	The ID of the video to delete
sSessionId	User session id in guid format
sUserName	Netlinx generated username

Sample XML

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
</v2>
```

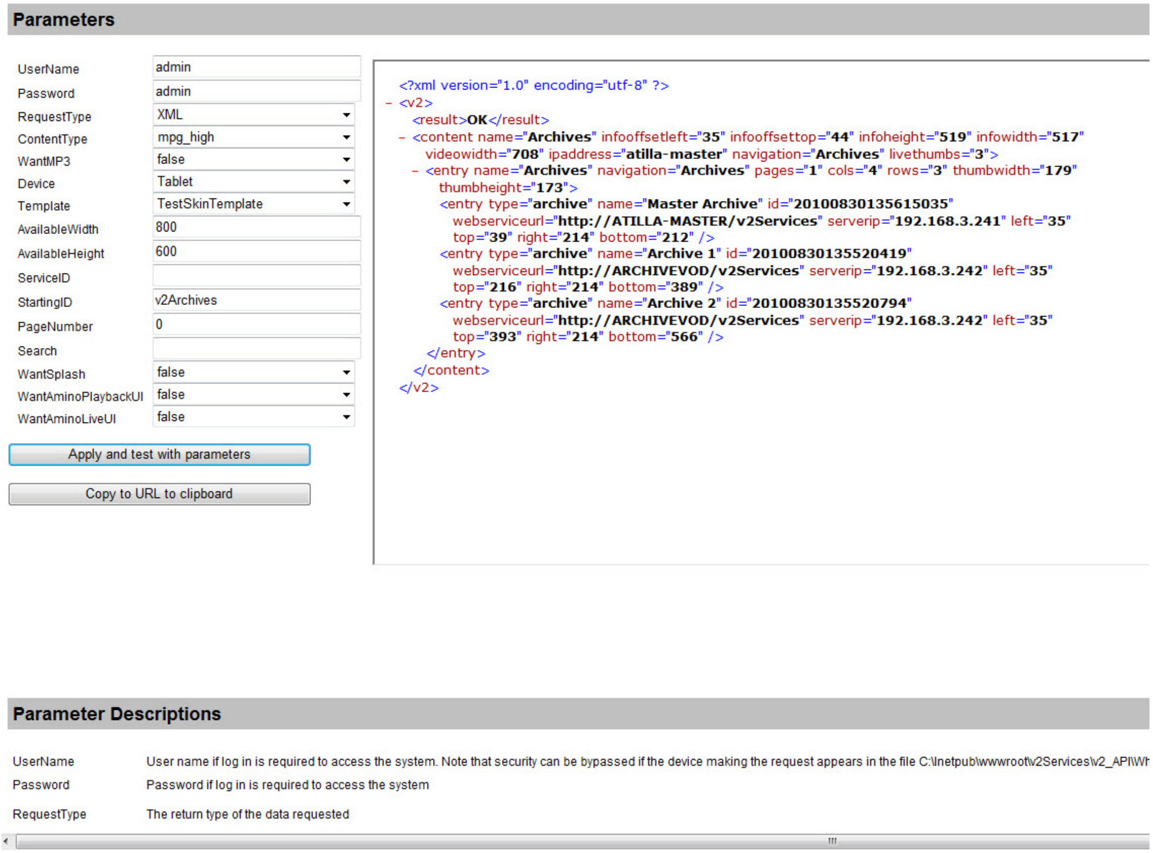
Vision^2 RenderThumbs Explorer

The Vision^2 RenderThumbs Explorer is a Meta API helper for creating any user interface which needs to provide the user with a graphical way to navigate or search the Vision^2 Archives on the system. Underneath the hood it is making calls to new Vision^2 API but this method makes creating user interfaces much easier than calling the raw API directly.

The test page on the Vision^2 System at *http:/_v2server_name_/v2Services/v2_API/RenderTest.aspx* can be used to interactively explore how the API works.

The way to use this API is to make the same call twice, the first time using RequestType=XML or RequestType=JSON and then the second time using RequestType=JPEG.

FIG. 1 displays an XML call:.



Parameters

UserName	admin
Password	admin
RequestType	XML
ContentType	mpg_high
WantMP3	false
Device	Tablet
Template	TestSkinTemplate
AvailableWidth	800
AvailableHeight	600
ServiceID	
StartingID	v2Archives
PageNumber	0
Search	
WantSplash	false
WantAminoPlaybackUI	false
WantAminoLiveUI	false

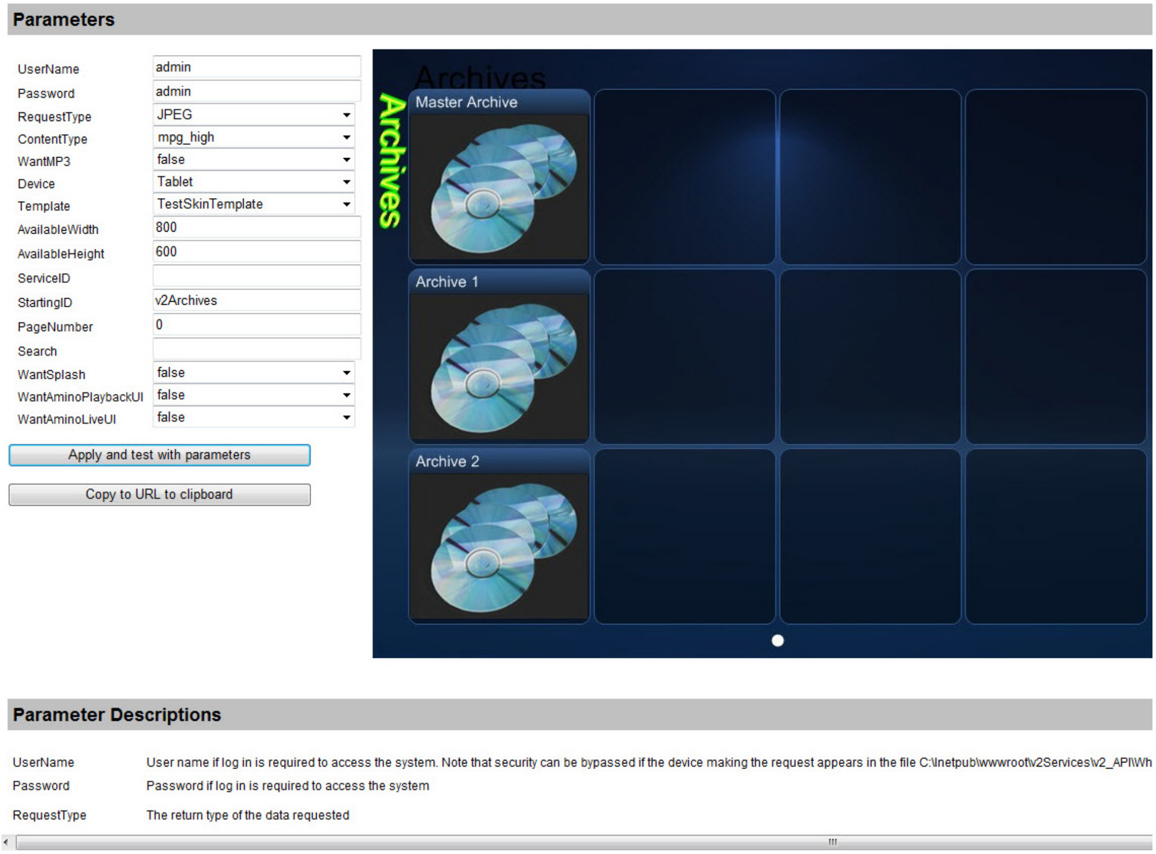
```
  <?xml version="1.0" encoding="utf-8" ?>
- <v2>
    <result>OK</result>
  - <content name="Archives" infooffsetleft="35" infooffsettop="44" infoheight="519" infowidth="517"
      videowidth="708" ipaddress="atilla-master" navigation="Archives" livethumbs="3">
    - <entry name="Archives" navigation="Archives" pages="1" cols="4" rows="3" thumbwidth="179"
        thumbheight="173">
        <entry type="archive" name="Master Archive" id="20100830135615035"
          webserviceurl="http://ATILLA-MASTER/v2Services" serverip="192.168.3.241" left="35"
          top="39" right="214" bottom="212" />
        <entry type="archive" name="Archive 1" id="20100830135520419"
          webserviceurl="http://ARCHIVEVOD/v2Services" serverip="192.168.3.242" left="35"
          top="216" right="214" bottom="389" />
        <entry type="archive" name="Archive 2" id="20100830135520794"
          webserviceurl="http://ARCHIVEVOD/v2Services" serverip="192.168.3.242" left="35"
          top="393" right="214" bottom="566" />
      </entry>
    </content>
  </v2>
```

Parameter Descriptions

UserName	User name if log in is required to access the system. Note that security can be bypassed if the device making the request appears in the file C:\Inetpub\wwwroot\v2Services\v2_API\Wh
Password	Password if log in is required to access the system
RequestType	The return type of the data requested

FIG. 1 Vision^2 RenderThumbs Explorer - XML

FIG. 2 displays a JPEG call:.



Parameters

UserName	admin
Password	admin
RequestType	JPEG
ContentType	mpg_high
WantMP3	false
Device	Tablet
Template	TestSkinTemplate
AvailableWidth	800
AvailableHeight	600
ServiceID	
StartingID	v2Archives
PageNumber	0
Search	
WantSplash	false
WantAminoPlaybackUI	false
WantAminoLiveUI	false

Parameter Descriptions

UserName User name if log in is required to access the system. Note that security can be bypassed if the device making the request appears in the file C:\Inetpub\wwwroot\v2Services\v2_API\Wh

Password Password if log in is required to access the system

RequestType The return type of the data requested

FIG. 2 Vision^2 RenderThumbs Explorer - JPEG

Initial Requests

Using the default settings when the page loaded, change the *RequestType* parameter to *XML* and then click Apply and test with parameters. If dealing with multiple Archives, the returned XML will be seen then in the right hand area and should resemble these below.

Note: *If your system does not have more than one Archive, then the call will not return a single Archive but will automatically return the content for the only Archive on the system (see next XML section for sample results)*

The actual call the test page makes will be similar to the following URL:

```
http://_Server_name_/v2Services/v2_API/v2_RenderThumbs.aspx?UserName=&Password=
&RequestType=XML&ContentType=mpg_high&Device=Tablet&WantMP3=false
&Template=720x576_4x4_blue&AvailableWidth=800&AvailableHeight=600
&ServiceID=&StartingID=v2Archives&PageNumber=0&Search=&WantSplash=false
&WantAminoPlaybackUI=false&WantAminoLiveUI=false
```

The following table lists the parameter descriptions for the above URL. These definitions also appear on the Vision^2 RenderThumbs Explorer screen.

Parameter Descriptions	
UserName	User name if log in is required to access the system. Note that security can be bypassed if the device making the request appears in the file: C:\Inetpub\wwwroot\v2Services\v2_API\WhiteList.xml
Password	Password if log in is required to access the system
RequestType	The return type of the data requested
ContentType	The format and bitrate of videos required
Device	The device being used to play the video. This changes the video URLs returned by the call to be correct for the type of device. For example the Amino uses RTSP while Tablets use HTTP calls.
Template	The template defines how the thumbnails are laid out and the colors used. Templates are stored on the system under: C:\Inetpub\wwwroot\v2Services\UI\Generic\Templates
AvailableWidth	The width available on the target device for the UI.
AvailableHeight	The height available on the target device for the UI.

Parameter Descriptions	
ServiceID	The Service ID of the Archive from which to return the data. The available IDs are returned by leaving this entry blank and requesting a StartingID of "v2Archives".
StartingID	The ID from which to return the data. If "v2Archives" is requested a list of all of the archives (and their IDs) on all systems will be seen (note that if there is only one archive on the entire system then the root content for the only archive back will be seen.) If "VideoCast_ContentRoot" or "" are requested, then the root content of the archive server made the v2_RenderThumbs call to will be seen. If the ID is a category, a list/image of thumbnails will be returned. If the ID is an actual video, a background for a information page (if RequestType=JPEG) will be returned, or the Metadata for that video (if RequestType=XML) will be returned. Note that on this page Internet Explorer will display the XML with the &lt; and &gt; entries automatically converted to < and >. Perform these replacements if actual HTML is desired. (Right click on the XML on this page and select View Source and view what is actually returned.)
PageNumber	When the number of thumbnails exceeds the total available as defined by the template then this parameter allows selecting which page of thumbs are returned. Note that this parameter is only valid when requesting an image as when requesting XML or JSON then the entries for all of the content at the specified ID will be returned.
Search	If this parameter is not empty then it overrides the StartingID. If the search string contains more than one word then the exact string is searched for. The search string can also contain the terms AND a NOT. For example searching for Kylie AND Jason will find videos which contain both words. Searching for Kylie NOT Jason will find videos which contain the word Kylie but do not contain the word Jason.
WantSplash	If this is true then a Splash Image is created. This is built from the background image from the template folder plus a PNG file from the same folder called Splash_ + Device + ".png".
WantAminoPlaybackUI	Custom parameter for Amino GUI only.
WantAminoLiveUI	Custom parameter for Amino GUI only.

In the following descriptions, all the test page is doing is changing these parameters and then re-submitting the call to the server.

Note: *The call automatically returns the content even if the system has only one Archive (see the next XML section for sample results).*

```
<?xml version="1.0" encoding="utf-8" ?>
<v2>
  <result>OK</result>
  <content
          name="Archives"
          infooffsetleft="20" infooffsettop="44" infoheight="519" infowidth="547"
          videowidth="746"
          ipaddress="43.192.168.100"
          navigation="Archives"
          livethumbs="2">
      <entry
             name="Archives"
             navigation="Archives"
             pages="1"
             cols="3" rows="3"
             thumbwidth="252" thumbheight="173">
          <entry
              type="archive"
              name="Archive 1"
              id="cc4597b2-e336-424f-bcca-a728c1e0c5d9"
              webserviceurl="http://WIN2K3SERVER/v2Services"
              serverip="43.192.168.100"
              left="20" top="39" right="272" bottom="212" />
          <entry
              type="archive"
              name="Archive 1 on Lenovo"
              d="7afc1b35-2d3a-49b1-8c86-5b84eca25eab"
              webserviceurl="http://LENOVO/v2Services"
              serverip="43.192.168.201"
              left="20" top="216" right="272" bottom="389" />
    </entry>
  </content>
</v2>
```

Initially, the first thing to look for is that the result is OK. If the result is not OK then an error message will be seen.

Note: *If multiple entry tags with a type labeled "archive" are seen, then there are multiple archives on the system.*

Return to the options on the RenderThumbs Explorer page, change the RequestType option to **JPEG**, and click **Apply and test with parameters** again. Vision^2 renders a user interface according to the template indicated on the RenderThumbs Explorer page. The result should appear similar to the image in FIG. 3.

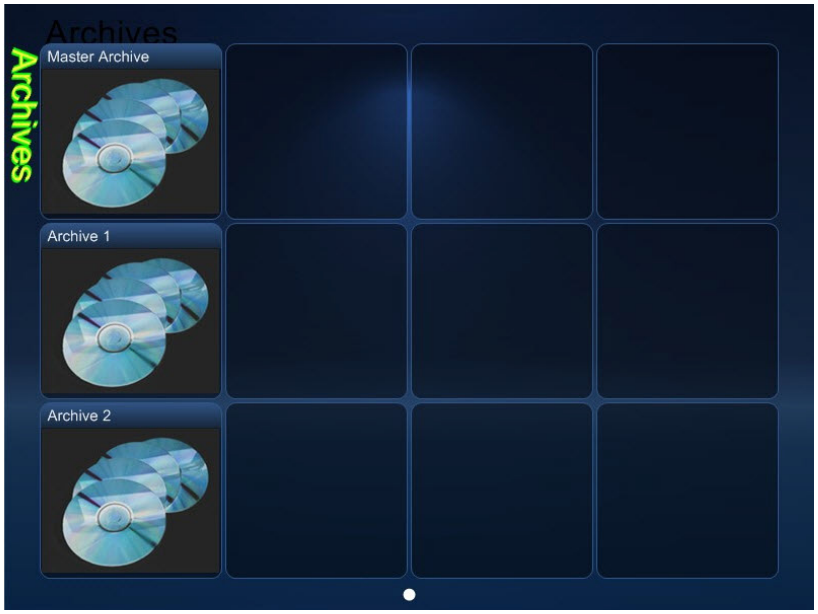


FIG. 3 Archive 3x3 grid

The following table lists the possible tags and attributes that may be seen in the XML code:

XML Tags and Attributes

Tag	Attribute	Description
content	livethumbs	Indicates how many of the thumbnails on the grid contain selectable entries.
Parent Entry Tag	pages	How many pages are required to display all of the available thumbnails
	cols	How many columns are in the grid
	rows	How many rows are in the grid
Entry	type	The type of thumbnail - can be 'archive', 'folder', 'programme' or 'search'
	id	The ID of the entry
	Left	The left offset to the start of this thumbnail (in pixels)
	Top	The top offset to the start of this thumbnail (in pixels)
	Right	The right offset to the start of this thumbnail (in pixels)
	Bottom	The bottom offset to the start of this thumbnail (in pixels)

For each thumbnail entry, determine where on the image the thumbnail is located and the id of that thumbnail. In the example XML, the user clicks on the image, and the x- and y-coordinates fall between x>=20&& y>=29&&x<=272&y<=212, then the user clicked on the first thumbnail which has an ID of 20110014111919390.

On the RenderThumbs Explorer page, change the RequestType back to XML and click **Apply and test with parameters**. Next, copy the ID of the first thumbnail into the StartingID parameter. Finally, edit the ServiceID parameter so it is empty and click **Apply and test with parameters**. Updated XML will be displayed as below:

```
<v2>
  <result>OK</result>
   <content
        name="Archive 1"
        contentarchive="/Archives/Archive1"
        physicalpath="H:\vcContent\Archive1\"
        infooffsetleft="20" infooffsettop="44" infoheight="519" infowidth="547"
        videowidth="746"
        ipaddress="localhost"
        navigation="Content"
        livethumbs="5">
        <entry
            type="category" name="Content" id="VideoCast_ContentRoot"
```

```
          navigation="Content"
          basecontenturl="http://WIN2K3SERVER/v2Services/Archives/Archive1"
          pages="1"
           cols="3" rows="3"
          thumbwidth="252" thumbheight="173">
         <entry
             type="search" name="Search" id="Search"
             contentpath=""
             left="20" top="39" right="272" bottom="212" />
         <entry
             type="category"
             name="Automatic Recordings"
             id="VideoCast_DVRAutoRecordings"
             contentpath="/1500/73/"
             left="20" top="216" right="272" bottom="389" />
         <entry
             type="category" name="All Music Videos" id="2011016114162"
             contentpath="/1500/25/"
             left="20" top="393" right="272" bottom="566" />
         <entry
             type="category" name="Music by artist" id="20107397906"
             contentpath="/1500/25/"
             left="274" top="39" right="526" bottom="212" />
         <entry
             type="category" name="HD Content" id="20107241343953"
             contentpath="/0000/77/"
             left="528"
             top="39"
             right="780" bottom="212" />
</entry>
</content>
```

The values of interest returned by the XML are again the same as before. This time, however, the entry types for the thumbnails are either ’category’ or ’search’. When a user selects the search thumbnail, it must be handled slightly differently from any other thumbnail (see the *Handling Search Thumbnails* section on page 26 for more information.)

On the RenderThumbs Explorer page, change the RequestType back to JPEG and click **Apply and test with parameters**. The image for the current selection will be seen (FIG. 4).

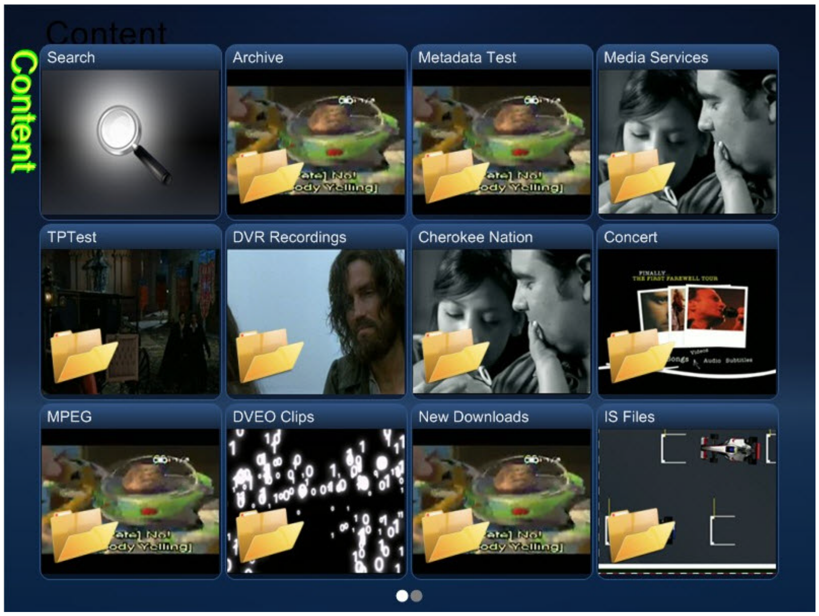
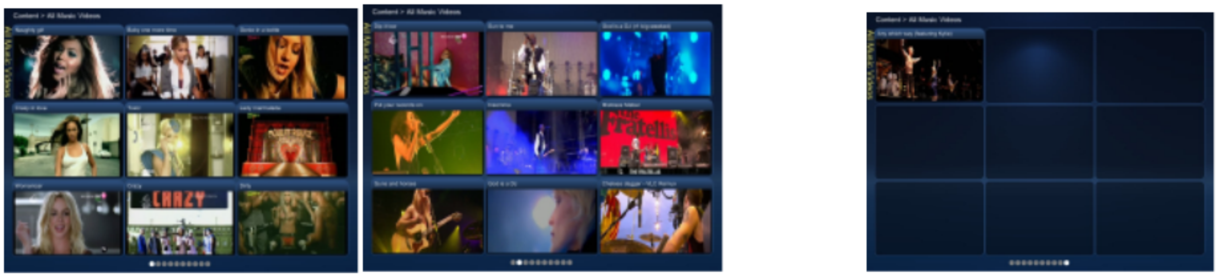


FIG. 4 Content page

To provide navigation down through the folders, change the value of the StartingID parameter to the ID of the thumbnail, and resubmit the two calls sequentially when a user clicks on a thumbnail.

At this point, it is worth considering the pages value of the parent entry node. When requesting the XML (or JSON) version, the data contains the values for every thumbnail in that category. However, when requesting the JPEG image, the image only contains the number of columns and rows defined by the selected template. If the number of thumbs displayed is less than the number that exists in the category, a means to navigate to the previous and next pages must be provided to the user. To obtain a new page, simply change the PageNumber parameter and resubmit the request for the JPEG.



PageNumber=0 PageNumber=1 PageNumber=9

FIG. 5 Content page

When thumbnail entry type is 'programme' rather than 'category', then both the XML and JPEG returned are different. The XML returned will contain the metadata for the video and the JPEG will be formatted to provide a background on which to display the metadata.

For example the XML will look similar to:

```
<?xml version="1.0" encoding="utf-8"?>
<v2><result>OK</result>
<info>&lt;p class='v2InfoSectionHeading'&gt;Synopsis&lt;/p&gt;
 &lt;p class='v2InfoSection'&gt;Naughty Girl is an R&amp;amp;Bdisco song written by Beyoncé Knowles,
Scott Storch, Robert Waller, and Angela Beyinc for Beyonc's debut solo album, Dangerously in Love
(2003). Produced by Beyoncé and Storch, the song lives up to the Western sound of its predecessor "Baby
Boy". Along with its several awards garnered, the song received positive responses from critics, citing
its sensual content. &lt;br/&gt;&lt;br/&gt;The single was released as the fourth and final single from
the album early in 2004. Though it failed to match the success of "Crazy in Love" and "Baby Boy", however
still gaining enormous success reaching number three on the U.S. Billboard Hot 100, the single had
immediate success which helped the album propel on charts. The single received similar responses from
international music markets, entering mostly top twenty. The single's music video features Beyoncé
flirtatiously and seductively dancing with Atlanta singer Usher to portray a naughty girl. The video
gained her an additional award.&lt;br/&gt;&lt;/p&gt;</info></v2>
```

Note: *When this code is viewed in the RenderThumbs Explorer page, Internet Explorer displays the XML with the &lt; and &gt; entries of the info node automatically converted to < and >. If actual HTML is desired, perform these replacements manually. When performed correctly, valid HTML will be obtained.*

Handling Search Thumbnails

When a user clicks on a thumbnail which has an ID of 'search', the user interface must execute a search. To execute a search, the Search parameter must change to contain the search string. Leave all other parameters as is. When finished, resubmit the requests. Both XML code and JPEGs will be obtained as if the result of the search was just another folder.

If the search string contains more than one word, the search is performed for the exact string of words. The search string can also contain the terms AND and NOT. For example, searching for Kylie AND Jason finds videos which contain both words. Searching for Kylie NOT Jason finds videos which contain the word Kylie but do not contain the word Jason.

Note: *After a search has returned its results for both the XML and JPEG requests, the Search parameter must be set back to being an empty string before calling it again with a new StartingID.*

Playing a Video

A Video ID parameter correctly formatted for the selected Device is available in the videopath attribute of a thumbnail entry node for each video.

AMX does not assume responsibility for errors or omissions. AMX also reserves the right to alter specifications without prior notice at any time.

The AMX Warranty and Return Policy and related documents can be viewed/downloaded at www.amx.com.

3000 RESEARCH DRIVE, RICHARDSON, TX 75082
AMX.com | 800.222.0193 | 469.624.8000 | +1.469.624.7400 | fax 469.624.7153

Last Revised:
2/01/2016