1 Introduction

This document describes the interface between ContentDepot and Station Automation Systems. The processes for scheduling, delivering and broadcasting programs will operate in the following sequence

1. Stations will subscribe to programs on ContentDepot
2. Stations will schedule the programs for transmission
3. Programs will be delivered to the stations
4. Stations will broadcast the programs

Each of steps 3 to 4 above will involve interfaces to the station automation system, which may be executed either by user interaction or by a machine-to-machine exchange of data, according to the capabilities of the automation system. All stations will have at least a basic station automation system where some of the interfaces will be manual while some stations may develop their systems to automate those functions. This document notes where a machine-to-machine interface is not supported by the basic automation.

The objective is to provide stations with a reliable end-to-end system that requires the minimum of user input to achieve the required output commensurate with their investment.

Each process is described below by means of one or more sequence diagram(s) and, where data is exchanged, a data definition. The sequence diagrams show options where a process may be automated or manual, depending on the capabilities of the station automation system. For the delivery to stations, separate sequence diagrams show stream delivery and file delivery.
1.1 End to end Process

The diagram below shows a high level view of the end-to-end process

1. Provide Program Schedule Metadata (Id, Title, Schedule)
2. Create Assets
3. Create Station Schedule
4. Deliver File
5. Link Content to Database Record (files only)
6. Execute Schedule
7. Play Files
8. Deliver Stream
9. Broadcast Content
10. Unsubscribe to Program
11. Delete schedule and files

1.2 Glossary

Program – The framework for a series of episodes. A program comprises 1 or more episodes
Program Title – A free text title for the program (e.g. Morning Edition)
Program UID – A short title for the program that is unique in ContentDepot (e.g. MornEd)
Episode – A single instance of an editorial entity for broadcast. An episode comprises 1 or more segments.
Segment – The actual essence that is broadcast. Segments are concatenated.
File Name – Name of the file. It is constructed from Program UID + Episode number + Segment number
Cutnum – The unique identifier for a segment.
Basic Automation System – the minimum station automation system capabilities necessary to enable a station to play files according to a station schedule.
Floating Cut-away – A period of predetermined duration that starts within a predetermined window of time that is normally used to enable stations to insert local material into a national program.
Stream – A feed of program material over the satellite link from NOC in real time.
NOC – Network Operations Centre
2 Processes and Interfaces

2.1 Create Assets and Station Program Schedule

2.1.1 Process
Station Operator will retrieve information from ContentDepot concerning programs they have subscribed to for the purposes of creating a station schedule in the Station Automation System. This process will also create “takes” or records in the station automation system database to which audio files will be linked on a one file per take basis.

A picture of the user interface to ContentDepot is shown for information.
The Station Operator will log onto ContentDepot and view the appropriate date (e.g., the time frame and program names). For each episode of a program, the user reads the information from ContentDepot. The user will enter the information for each file into the Station Automation System database and scheduling tool that will be received and played (to which the audio file can be attached later).

If the file is non-essence, the file will not be able to be scheduled until received. For essence content, the station operator looks up Cut IDs of segment files he wants to use in a play-list using CD Portal and creates empty cuts in SAS library with same IDs. This would only have to be done once per program or any time Content Producer decides to change number of segments for entire program or specific episode.
2.1.2 User Interface Data Elements

The user enters information via the Station Automation System GUI.
Required – Needed to enable the station automation system to create a valid station schedule
Optional – Additional information to be used by the Station Operator.

For each segment of an episode that is a file, the user creates a Database Record on the Station Automation System

<table>
<thead>
<tr>
<th>Data Element Name</th>
<th>Description</th>
<th>Required/Optional/Conditional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cutnum (for each segment)</td>
<td>Unique identifier of the segment</td>
<td>Required – will be provided by ContentDepot if episode is delivered as files and the file is broadcast essence.</td>
</tr>
<tr>
<td>Program Title</td>
<td>Program title</td>
<td>Optional</td>
</tr>
<tr>
<td>Genre</td>
<td>Talk, News, Music, etc.</td>
<td>Optional</td>
</tr>
<tr>
<td>Artist</td>
<td>Names of Artist(s)</td>
<td>Optional</td>
</tr>
</tbody>
</table>

For each broadcast of an episode, a schedule (or playlist) of segments is created on the Station Automation System

<table>
<thead>
<tr>
<th>Data Element Name</th>
<th>Description</th>
<th>Required/Optional/Conditional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scheduled Air Date</td>
<td>Date the episode will be broadcast.</td>
<td>Required</td>
</tr>
<tr>
<td>Scheduled Air Time</td>
<td>Time the episode broadcast will start.</td>
<td>Required</td>
</tr>
<tr>
<td>Segment type</td>
<td>Live stream or file</td>
<td>Required</td>
</tr>
<tr>
<td>Cutnum (for each segment)</td>
<td>The unique identifier of the segment</td>
<td>Required – will be provided by ContentDepot if episode is delivered as files and the file is broadcast essence.</td>
</tr>
<tr>
<td>Decoder port (audio output number)</td>
<td>The ContentDepot source for live streams</td>
<td>Conditional - Required if any segment is a live stream</td>
</tr>
<tr>
<td>Program Title</td>
<td>Program title</td>
<td>Optional</td>
</tr>
<tr>
<td>Genre</td>
<td>Talk, News, Music, etc.</td>
<td>Optional</td>
</tr>
<tr>
<td>Artist</td>
<td>Optional</td>
<td></td>
</tr>
</tbody>
</table>

Note: The cutnum provided by ContentDepot will reset based upon the frequency of a program. For example:

- Daily show with 6 segment files for each episode [pool 80200-80241]:
  - Monday – 80200, 80201, 80202, 80203, 80204, 80205
  - Tuesday – 80206, 80207, 80208, 80209, 80210, 80211
  - Wednesday – 80212, 80213, 80214, 80215, 80216, 80217
  - Thursday – 80218, 80219, 80220, 80221, 80222, 80223
  - Friday – 80224, 80225, 80226, 80227, 80228, 80229
  - Saturday – 80230, 80231, 80232, 80233, 80234, 80235
  - Sunday – 80236, 80237, 80238, 80239, 80240, 80241

- Weekly show with 5 segments for each episode [pool 80300-80304]
  - Tuesday – 80300, 80301, 80302, 80303, 80304

Note: Please refer to “Technical specification of interface between Content Depot Delivery Subsystem and Station Automation System” version 1.2 for details on how cut numbers are assigned.
2.2 Deliver Content File

2.2.1 Process

ContentDepot will send the program files to stations that have subscribed to or requested a download of subscribed programs. All content files [both essence and non-essence] get delivered to Storage Receiver and placed under so called <archive> directory and are available to be manually pulled to SAS.

For essence files [segment files] whose delivery to Storage Receiver is triggered by producer uploading content to CD portal are exposed as symbolic links in designated directories <CDLongname> and CDCutID to be automatically pulled to SAS. SAS interrogates given station’s mapping and either uses a cut ID determined by CD and stored in the cart chunk under “cutnum” or assigns new one based on station mapping. The Station Automation System uses the cutnum to link the correct file to a database record. The Station Automation System will extract the metadata and update its database as necessary.

The Station Automation System should check the metadata and warn the operator if a file is scheduled to be broadcast before the StartDate and StartTime or after the EndDate and EndTime have been reached. The station automation system should purge files from its storage 1 day after the EndDate has been reached.
2.2.2 Interface – Machine Interface

Files will conform to the broadcast wave files (.wav) specification and include the following chunks.

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fmt Chunk</td>
<td>As defined by Microsoft</td>
</tr>
<tr>
<td>Fact Chunk</td>
<td>As defined by Microsoft</td>
</tr>
</tbody>
</table>

**Bext** - If a source file contains metadata in a bext chunk to the CartChunk/AES46-2002 standard, this will be included in the broadcast wave file (.wav)

<table>
<thead>
<tr>
<th>Bext Chunk</th>
<th>ContentDepot Attribute</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>None</td>
<td>Description of the sound sequence CHAR [256]</td>
<td>NULL</td>
</tr>
<tr>
<td>Originator</td>
<td>None</td>
<td>Name of the originator CHAR [32]</td>
<td>NULL</td>
</tr>
<tr>
<td>OriginatorReference</td>
<td>None</td>
<td>Reference of the originator CHAR [32]</td>
<td>NULL</td>
</tr>
<tr>
<td>OriginationDate</td>
<td>None</td>
<td>The date of creation of the audio sequence. CHAR [10] yyyy.mm.dd</td>
<td></td>
</tr>
<tr>
<td>OriginationTime</td>
<td>None</td>
<td>The time of creation of the audio sequence. CHAR [8] hh:mm:ss</td>
<td></td>
</tr>
<tr>
<td>TimeReferenceLow</td>
<td>None</td>
<td>This field contains the timecode of the sequence. It is the first sample count since midnight. DWORD</td>
<td>0000</td>
</tr>
<tr>
<td>TimeReferenceHigh</td>
<td>None</td>
<td>DWORD</td>
<td>0000</td>
</tr>
<tr>
<td>Version</td>
<td>None</td>
<td>Version of the BWF binary number</td>
<td>0001h</td>
</tr>
<tr>
<td>Binary byte 0 of SMPTE Umid</td>
<td>None</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Binary byte 63 of SMPTE Umid</td>
<td>None</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Reserved</td>
<td>Reserved for future use</td>
<td>BYTES [190]</td>
<td>NULL</td>
</tr>
<tr>
<td>CodingHistory</td>
<td>None</td>
<td>Populated by ContentDepot to identify the type of file. The &quot;A&quot; (file format), &quot;F&quot; (sampling rate) and &quot;B&quot; (bit rate) parameter values will be populated from the values in the mpeg header of the audio essence file. ASCII</td>
<td>Will follow the following format where XXX are values: A=XXX,F=XXX,B=XXX,M=XXX,T=XXX+char13+char10 Example: A=MPEG1L2,F=44100,B=256,M=STEREO,T=CV_PcxT12NP&quot; + char13 + char10.</td>
</tr>
</tbody>
</table>
Cart - Some fields of the CartChunk/AES46-2002 map to attributes within ContentDepot. These attributes will be sent in the CartChunk/AES46-2002 of the broadcast wave file (.wav).

<table>
<thead>
<tr>
<th>CartChunk/AES46-2002</th>
<th>ContentDepot Attribute</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Title</td>
<td>Segment title (Most automation systems display at least sixteen characters, usually at least 32 characters.)</td>
<td>Construct from program title, episode number, segment number, concatenated and truncated to fit</td>
</tr>
<tr>
<td>Artist</td>
<td>Hosts</td>
<td>The names of the Program host(s)</td>
<td>Text (64)</td>
</tr>
<tr>
<td>Cutnum</td>
<td>Cutnum</td>
<td>Unique identifier for the essence</td>
<td>For non-essence: Will be blank For essence: A 5 digit numeric field constructed from a pool of pre-defined numbers based upon the frequency of the program (daily, weekly, monthly)</td>
</tr>
<tr>
<td>Client ID</td>
<td>None</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>None</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>Classification</td>
<td>None</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>Out Cue</td>
<td>None</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>Start Date</td>
<td>Release Date</td>
<td>Date program available for air</td>
<td>YYYY-MM-DD</td>
</tr>
<tr>
<td>Start Time</td>
<td>Release Time</td>
<td>Time program available for air</td>
<td>hh:mm:ss</td>
</tr>
<tr>
<td>End Date</td>
<td>End Air Date</td>
<td>Date after which program should no longer be aired</td>
<td>YYYY-MM-DD</td>
</tr>
<tr>
<td>End Time</td>
<td>End Air Time</td>
<td>Time on the appointed date after which program should no longer be aired</td>
<td>hh:mm:ss</td>
</tr>
<tr>
<td>ProducerAppID</td>
<td>None</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>ProducerAppVersion</td>
<td>None</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>UserDef</td>
<td>None</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>DwLevelReference</td>
<td>None</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>PostTimer</td>
<td>None</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>URL</td>
<td>None</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>TagText</td>
<td>None</td>
<td>Text seen by the public on the radio display. Will contain a static value for all files and will be composed of the following 8 fields: OwnerIdentifier=&quot;ContentDepot&quot; Identifier=&quot;001&quot; Title=&quot;Talk about ContentDepot&quot; Artist=&quot;PRSS&quot; Album=&quot;ContentDepot Test Program&quot; Genre=&quot;101&quot; CommentTitle=&quot;This is a comment&quot; Comment=&quot;These are test and placeholder fields for ContentDepot PAD support in files. They are currently generated from static fields in the ContentDepot Portal database. Send email to <a href="mailto:prssplanning@npr.org">prssplanning@npr.org</a> if you have questions.&quot; Note: This sample data will be provided during ContentDepot rollout. Program specific data will be available at a later date.</td>
<td></td>
</tr>
</tbody>
</table>

Detailed rules on the naming of the file:
- Will be constructed using the following: Program UID_Episode number_category+sequential number (8+1+3+1+4+2=19 characters).
- **PROGRAM_UID**: Use first 8 non space characters of program title or contentexchange id for program id. If this is not unique, use the first 6 non space characters and a 2 digit sequential number.
- **Category**: PROM - Promo, EVGN - Evergreen, WEBM - Web Module, IMAG for Image, PHOTO for Photo, LOGO for Logo, SGMT - Segment, EXCH - Content Exchange (will include contentexchange id, not program id), TEMP - Program Template, MAIL - Mail, RUDN - Rundown, PAD - Pad, TEXT - Text.
Episode number: Will be a sequential number based upon the time it is created in ContentDepot (the first episode created gets 001 etc.). Non-episode content will use 000 for the episode number.

Sequential number: Will be a sequential number assigned by the system based upon the order in which the content was created (e.g., the first promo for an episode is episode 1, the first segment created for and episode is episode 2). This number starts at 01 and goes to 99 for each type of content for a particular program or episode (e.g., content types include: episode segments, episode promos, episode webmodules, program webmodules). The numbering resets to 01 after 99.

Tag Text will be provided in the following XML Format:

```xml
<meta-data>
<property name="OwnerIdentifier" value="ContentDepot"/>
<property name="Identifier value="001"/>
<property name="Title" value="Talk about ContentDepot"/>
<property name="Artist" value="PRSS"/>
<property name="Album" value="ContentDepot Test Program"/>
<property name="Genre" value="101"/>
<property name="CommentTitle" value="This is a comment"/>
<property name="Comment" value="These are test and placeholder fields for ContentDepot PAD support in files. They are currently generated from static fields in the ContentDepot Portal database. Send email to prssplanning@npr.org if you have questions."/>
</meta-data>
```

Wave - The audio data will be coded to MPEG 1 Layer 2, 44,100 samples per sec, 2 channels (stereo), bit reduced to 256kb/sec stereo.

Files delivered to the station receiver will be provided in the following directory structure:

```xml
<Receiver Host Name>
  \<archive>
    \PROGx_EpisodeY_Seg1>
      ... And so forth...
  \<CDLongname>
    \<symbolik_link_to_PROGx_EpisodeY_Seg1>
      ...And so forth...
  \<CDCutID>
    \<symbolik_link_to_PROGx_EpisodeY_Seg1>
      ...And so forth...
CDLongname link names are identical to the file names in <archive>. CDCutID link names are equal to cut IDs nominated by the CD Delivery system.
```

The <CDCutID> directory will have symbolic links created using the same rules as the <CDLongName> except the name of the file will match the cutnum field. This directory is optionally available for SAS systems to pull content files into the SAS cut library based on the file name instead of the cutnum metadata field.

### 2.2.3 Replace Segment File

For essence delivered as part of a subscription (ContentDepot pushes the file to the station receiver), the SAS must determine if the file is a newer version of the file or an older version than what is available in the SAS cut library.

If no Cutnum matches in the SAS cut library:

- The SAS system will import the file into the SAS cut library

If a cutnum matches a file in the SAS cut library:

- A) If the last modified date of the new file is older than the last modified date of the existing file in the SAS cut library, the SAS system will not import the file.
- B) If the last modified date of the new file is newer than the last modified date of the existing file in the SAS cut library, the SAS system compares the StartDate of the new file:
  - a. If the StartDate of the new file is newer/more recent than the StartDate of the existing file in the SAS cut library, the SAS system replaces the file
  - b. If the StartDate of the new file is older/before the StartDate of the existing file in the SAS cut library, the SAS system ignores the new file

For non-essence or essence requested by the station:

If a file is received with the same file name as an existing one in the station automation system store, the existing file is replaced with the later one. It will be the responsibility of the user to import the file into the station automation database and place it in the desired playlist.
2.3 Broadcast Stream

2.3.1 Process
This is the routing of a stream for the purposes of live broadcasting or a replay from the NOC in real time.

2.3.2 Option 1: Manual
The Station Operator is responsible for switching the audio output of the stream to the appropriate station broadcast audio chain for the purpose of broadcasting.

2.3.3 Interface – User Interface
There is no machine – machine interface. The user operating the appropriate controls on the station automation system and broadcast system achieves the functionality
2.3.4 Option 2: Automated.

The Station Automation System is responsible for switching the audio output of the stream to the appropriate station broadcast audio chain for the purpose of broadcasting.

2.3.5 Interface – Machine Interface

The Station Automation system will control the local station routing equipment to switch the program source to the correct IDC decoder audio output according to the station schedule.
2.4 Play Broadcast File

2.4.1 Option 1: Manual

The Operator switches the routing in the broadcast system and plays files to air

2.4.2 Interface – User Interface

The interface is provided by the Station Automation System GUI and broadcast system controls.
2.4.3 Option 2: Automated

The Station Automation controls the broadcast system routing and plays files to air according to the station schedule.

2.4.4 Interface – Machine Interface

The Station Automation system will control the local station routing equipment to switch the program source to the file playback audio output according to the station schedule.
2.5 Deliver Broadcast Stream Cues

Each audio stream on the ContentDepot supports 32 program-associated cues. Cues 1-4 are delivered via hardware relays associated with physical audio outputs on the back of the receiver. Cues 1-32 are delivered via XML-formatted data on the receiver’s Ethernet interface. Note that cues 1-4 are delivered via both hardware (relays) and via Ethernet to provide backwards compatibility for legacy systems at stations.

Broadcast stream cues are delivered to signal the beginning and end of programs (Cues #1 and #4, respectively), and to signal producer-defined cutaway or break points in a program (Cues #2, #3, and #5-32) A cutaway is a broadcast of local material that may be live or the playback of a file. At the end of the cutaway or break, stations usually re-join the national program on the stream. Stations may choose to ignore the cutaway in which case they will broadcast sustaining program sent on the stream.

In many cases, station staffers elect to perform cutaways or breaks manually based on either audible cues such as “This is NPR. . “ or at specific times in the live program.

2.5.1 Process

On receipt of a cue, the station optionally broadcasts a local insert for a pre-determined duration and then re-joins the live stream. The producer may choose which one of 30 cut-away signals to use during the live transmission, each having semantics defined by the producer. The local insert may be live announcement, a file, delivery of non-audio data on a downstream system, or some combination of all three.

2.5.2 Option 1 – Manual

The user plays a file.

2.5.3 Interface – User Interface

The interface is provided by the Station Automation System GUI.
2.5.4 Option 2 - Automated.

Depending on the floating cutaway cue signal sent by the producer, the Station Automation or other downstream system takes pre-defined action before re-joining the live stream.

Diagram:
1: Send Stream (includes cues, audio, and PAD)
2: Deliver Cue
3: Interpret Cue
4: Broadcast File (see Broadcast File - Automated)
2.5.5 Interface – Machine Interface (Hardware Relay Interface)

ContentDepot receivers activate contact closures based upon the cueing information provided within the audio stream. Nominal dwell time is 250 msec. Each receiver supports 4 stereo audio channels each with 4 associated relay closures that can be connected to the station automation system GPI’s.

The standard installation comprises 2 receivers. There are, therefore, 32 available relays in total, (see diagram in section 4). All closures will be momentary.

The hardware relays have consistent system-defined meanings:

<table>
<thead>
<tr>
<th>Cue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Program Start</td>
</tr>
<tr>
<td>2</td>
<td>#2 Closure</td>
</tr>
<tr>
<td>3</td>
<td>#3 Closure</td>
</tr>
<tr>
<td>4</td>
<td>End of Program</td>
</tr>
</tbody>
</table>

2.5.6 Interface – Machine Interface (Multicast UDP Interface)

ContentDepot receivers also send XML-formatted data cues via the receiver’s Ethernet interface. Each program stream can support up to 32 cues. The first four cues are delivered at the same time as the four hardware relays associated with a decoder output. In other words, a “program start” cue on a stream both activates a relay on a decoder and sends an XML-formatted message out the Ethernet interface.

The additional 28 XML-only cues (cues 5-32) are designed for use by producers to signal downstream systems for additional cutaways and breaks – including systems that aren’t in the direct broadcast chain like web stream encoders.

The XML-formatted cues are sent via UDP to an IP Multicast address and port defined by station administrators. Each cue has the following XML structure:

```
<cue>
  <decoder id="1">
    <stream>1</stream>
    <relay>3</relay>
    <action>On</action>
  </decoder>
  <decoder id="2">
    <stream>2</stream>
    <relay>3</relay>
    <action>On</action>
  </decoder>
</cue>
```

Where:
- id is the audio output of the receiver where the associated audio is being delivered – 1 through 4
- stream is the PRSS channel viewable through the ECG. The PRSS currently uses channels 1 through 20, and are often referred to as “S001” through “S020”.
- relay is the cue number. 1 through 32.
- action is either “Off” or “On”. By convention, each cue activation – for cues 1-32 -- consists of two messages: a “Cue On” message, then a “Cue Off” message following nominally 250 milliseconds later. This is to provide backwards compatibility with the legacy hardware relays. It’s anticipated that downstream systems using the XML data will ignore the “Off” messages and activate functions based solely on the “On” messages.
2.5.7 Unsubscribe to Content

2.5.8 Process
The Station Operator logs onto ContentDepot and unsubscribes to a program. The Station operator will search the station automation system database to find all segments that relate to all episodes of the program. The station operator deletes the database entries (note that ContentDepot will not request the station automation system to delete records or files). The station automation system will purge the files from its storage.

2.5.9 Interface – User Interface
ContentDepot only provides a user interface and means to unsubscribe on ContentDepot. There will be no interface to update the Station Automation System with this information.

2.6 Timecode
The station automation system will synchronize to an NTP service on the IDC storage receiver.
3 Exceptions
The section above notes where the basic automation system does not support a machine interface. This section describes other functions not supported by an interface between ContentDepot and the Station Automation systems.

3.1 Modify Program schedule
The user may make modifications to an existing program schedule in the station automation system. Changes will not be reported to ContentDepot and, therefore, ContentDepot will not be synchronized. If the user subsequently exports schedule information from ContentDepot, the schedule will be replaced. The user will consolidate any conflicts manually.

3.2 Broadcast File
ContentDepot will not directly instruct the station automation system to play a specific file. This process is solely the responsibility of the Station Automation System functionality through either the station broadcast schedule or a command from the operator.

3.3 Manage Station Resources
The adding and removing of station resources (e.g., IDC receivers), will be managed using the Station Resource interface. The integration of the station resource with the Station Automation System is solely the responsibility of the Station Automation System functionality.

3.4 Schedule Program Repeats
The information necessary to determine the time and date on the station schedule for repeats will not be captured in ContentDepot as part of the basic automation interface. The user must enter all scheduling information in the station automation system manually for program repeats. The station automation system should enable the operator to copy all of the information from one schedule event to another in a single transaction.

3.5 Record Stream for Playout
The ability to record streams to file is solely the responsibility of the station operator using the station automation system functionality if installed. The basic station automation system will not support recording.

3.6 Monitor Health of Station Automation
The Station Operator monitors the health of the Station Automation tool. This functionality is solely the responsibility of the Station Automation System.

3.7 Monitor Schedule for Missing Files
The Station Operator monitors the Station Automation Schedule for missing files. This functionality is solely the responsibility of the Station Automation System.

3.8 Missing File Alerts
The alerting of missing files by the Station Automation Tool, when a scheduled play-out is close (Configured by the Station Operator) to being executed. The ability to communicate the alert to ContentDepot is a manual process.

3.9 Force Station to broadcast a stream (breaking news)
ContentDepot will not signal stations or force the station broadcast plant to abandon the existing program and switch to an alternative stream.
4 Architecture

The following diagram shows the network architecture for a typical installation at the station. Some stations may have more receivers to simultaneously access more streams.

Each receiver supports 4 audio outputs, each comprising a balanced stereo analog audio pair, an AES3 stereo digital audio output 4 GPO relay outputs, and a Livewire-compatible IP output. Streams are delivered from the receivers in real time to the station broadcast plant.

Files are delivered to the receivers, each of which has at least 900 GB of local disk storage. The receivers operate on the Linux OS, and use Samba to emulate an NT share. The Station Automation system scans the receiver’s directory and import new files into its local storage. The station automation system should not import any file with the same filename and either the same or an earlier time and date stamp than one already existing in its storage.

It would be advantageous if the station automation system supported addresses for two receivers in its configuration, whereby it would normally poll one receiver and switch to the other in the case of no response.

Files are automatically purged from the storage receiver at a predetermined time.

There is no back channel for station automation systems to report receipt of files or errors to ContentDepot. Files that are either incomplete or corrupt will have a different file extension. The extension is changed in the IDC receiver when the file is ready for the station automation system to import it. Each file will be sent from the NOC to the receiver a fixed number of times to mitigate against any outages of the delivery system, although once a file has been successfully delivered and remains on the receiver’s storage, subsequent deliveries are ignored by the receiver.

---

Vendor_Automation_System_Integration_Interface_v1_5.doc
## Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Description</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>August 4, 2004</td>
<td>1.0</td>
<td>Initial Release</td>
<td>Charles Cook, Trevor Smith</td>
</tr>
<tr>
<td>August 19, 2004</td>
<td>1.1</td>
<td>Following first review, changes to 1.1 diagram 1.2 definition of Basic Automation System 2.1.4 process description 2.2.2 add audio coding standard 2.3.1 process description 2.5.5 remove reference to main &amp; reserve decoders 4 changes to architecture description</td>
<td>Trevor Smith</td>
</tr>
<tr>
<td>October 21, 2004</td>
<td>1.2</td>
<td>- Removed references to downloading a machine readable station schedule - Removed reference to a hierarchical scheduling tool so as not to infer this is part of the integration effort - Added details to the file naming scheme - replaced references to the cart chunk and AES46-2002 to “CartChunk/AES46-2002”</td>
<td>Charles Cook</td>
</tr>
<tr>
<td>December 1, 2004</td>
<td>1.3</td>
<td>- Included logic to assign cutnum for essence content - Added CodingHistory logic</td>
<td>Charles Cook</td>
</tr>
<tr>
<td>January 27, 2005</td>
<td>1.4</td>
<td>1. Section 1, Introduction, paragraph starting, &quot;Each of steps .. .&quot;: Change 'whilst' to 'while'. 2. Section 2.1.2, User Interface Data Elements, note at bottom of section: Add reference to technical specification (and version) for more detail on how cut numbers are generated. 3. Section, 2.2.1, Process, paragraph starting &quot;For essence files. .. .&quot;: change designated directory name to &lt;CDLongname&gt;. 4. Section 2.2.2, Interface -- Machine Interface, CartChunk table: Add PAD detail including XML format and note about sample data in TagText row of table. 5. Section 2.2.2, Interface -- Machine Interface, note at bottom: Add &lt;CDLongname&gt; location and receiver hostname (CDFile1 or CDFile2) to notes. 6. Section 2.2.2, Interface -- Machine Interface, notes at bottom: Remove C-style structure definition for CartChunk/AES46-2002. Unnecessary for this document and possibly confusing. 7. Section 2.2.3 Replace Segment File, &quot;For essence delivered. .. .&quot; The rules are confusing; it's unclear what the desired behavior of the SAS is supposed to be. Please recast. 8. Section 2.3.2 Option 1: Manual, process flow drawing: Remove the &quot;Station will have the option to tune manually&quot; block. CC: LIST PEOPLE: IS THIS A CORRECT DESCRIPTION OF THE STREAM DECODERS? IN OTHER WORDS, WILL STATION STAFF BE ABLE TO LOCALLY 'TUNE' DECODERS TO A PARTICULAR IP STREAM OR NOT? THIS IS A BIG QUESTION. PLEASE ADVISE.</td>
<td>Charles Cook</td>
</tr>
</tbody>
</table>
9. Section 2.3.4 Option 2: Automated, process flow drawing: Remove the "Station will have the option to tune manually" block. CC: LIST PEOPLE: IS THIS A CORRECT DESCRIPTION OF THE STREAM DECODERS? IN OTHER WORDS, WILL STATION STAFF BE ABLE TO LOCALLY 'TUNE' DECODERS TO A PARTICULAR IP STREAM OR NOT? THIS IS A BIG QUESTION. PLEASE ADVISE.

10. Section 2.4.1 Option 1: Manual, process flow: Change 'routeing' to 'routing'.

11. Section 2.4.3 Option 2: Automated, process flow: Change 'routeing' to 'routing'.

12. Section 2.5.5 Interface -- Machine Interface, description of cue system: What is the dwell time for contact closures? In other words, how long does a closure last? If it's a setup issue for the receivers, we need to refer to the stream decoder documentation.

13. Section 2.7 Timecode: Add 'storage' between 'IDC' and 'receiver'.

14. Section 3.7 Alert of Missing Files: change name in document to "Missing File Alerts".

15. Section 3.7 Missing File Alerts and Section 3.8 Monitor Schedule for Missing Files: reverse order in document. Proper order for the functions.

16. Section 4, Architecture, first and second paragraphs: Change 'multichannel decoder' to 'stream decoder' to bring into consistent usage.

17. Section 4, Architecture, second paragraph: Change 'analogue' to 'analog'.

18. Section 4, Architecture, third paragraph: Add 'at least 80 GB' between 'each of which has' and 'local disk storage'.

19. Section 4, Architecture, diagram: Remove 'DVB' from L-band input.

Technical Specification of Interface Between Content Depot Delivery Subsystem and Station Automation System, Version 1.1 3 December, 2004 (hereon known as "TSIBCDDSSAS", or 'tee-sib-cuddle-sass'): New Section, 2.5.6 Interface -- Multicast UDP Interface. Minor edits throughout document to correct and clarify.